Localisation et cartographie visuelles simultanées en milieu intérieur et en temps réel

Marion Decrouez

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Informatique et Mathématique Appliquée**

Arrêté ministériel : 7 août 2006

Présentée par

Marion Decrouez

Thèse dirigée par **James L. Crowley**

préparée au sein du **Laboratoire Informatique de Grenoble à l'INRIA Grenoble Rhône-Alpes**
et de **Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Modélisation des environnements dynamiques pour la localisation

Thèse soutenue publiquement le ,
devant le jury composé de :

M. Peter Sturm
Directeur de Recherche à l'INRIA Grenoble Rhône-Alpes, Président
M. Jean-Marc Lavest
Professeur à l'IUT de Clermont-Ferrand, Rapporteur
M. Michel Devy
Directeur de Recherche au LAAS-CNRS, Rapporteur
M. Cédric Demonceaux
Maître de conférences à l'IUT LE CREUSOT, Examinateur
M. François Gaspard
CEA-LIST Saclay, Examinateur
M. Romain Dupont
CEA-LIST Saclay, Examinateur
M. Frédéric Devernay
Chargé de recherche à l'INRIA Grenoble Rhône-Alpes, Examinateur
James L.Crowley
Professeur à l'INRIA Grenoble Rhône-Alpes, Directeur de thèse

Remerciements

Je tiens tout d'abord à remercier François Gaspard pour la confiance qu'il m'a accordée en me proposant de réaliser cette thèse au sein du laboratoire LVIC du CEA LIST et pour avoir su se rendre disponible tout au long de cette thèse malgré son agenda chargé. Merci à Romain Dupont, mon encadrant au LVIC, pour avoir su me prodiguer des conseils pertinents afin d'orienter mes recherches pendant ces trois années. Je remercie également James Crowley, mon directeur de thèse, pour m'avoir toujours fait confiance, me laissant conduire mes travaux à mon aise et pour m'avoir permis de travailler dans les meilleurs conditions.

Merci aux membres du LVIC et en particulier Steeve, Vincent, Mohamed et Jim pour toutes les discussions que nous avons eues, leur soutien et leur aide, ainsi que leur accueil sympathique lors de mes déplacements au CEA de Saclay. Au sein de l'équipe PRIMA de l'INRIA Grenoble, j'ai eu la chance de travailler avec de nombreuses personnes qui m'ont, chacune à leur façon, accompagné au cours de ces trois années. Je tiens particulièrement à remercier Antoine Meler pour ses conseils avisés et sa disponibilité et pour m'avoir épaulée tout au long de ma thèse. Merci à tous les membres de l'équipe PRIMA, d'hier et d'aujourd'hui, Thibault, Claudine, Sylvain, John, Rémi, Harsimrat, Lukas, Lucas, Matthieu, Dominique, Amaury, Eva, Pierre, Patrick, Adrien, Stan pour la bonne ambiance générale et pour avoir rendu le quotidien très agréable. Je remercie aussi sincèrement l'équipe pour s'être rendu disponible lors des diverses pré-soutenances et pour leurs conseils avisés.

J'aimerais remercier les secrétaires du CEA et de l'INRIA qui m'ont accompagnée dans les nombreuses démarches administratives que j'ai eu durant ces trois années, Elodie, Sandrine ainsi que Annie, Marion, Marie-Anne et Catherine.

La soutenance vient finaliser les trois années de travail de thèse. Je tiens à remercier chaleureusement les huit membres du jury de ma thèse. En particulier, Jean-Marc Lavest et Michel Devy pour avoir accepté de rapporter mes travaux. Je remercie également l'ensemble du jury pour les échanges très constructifs que nous avons eu au cours de la journée de soutenance. Enfin, un grand merci à Peter Sturm d'avoir présidé ma soutenance et d'avoir su rendre ce moment agréable.

Plus personnellement, je remercie mes parents et ma sœur Anne pour leur présence et leur soutien avant et pendant ma soutenance.

Je remercie Alex.

Table des matières

Résumé

Les travaux effectués dans cette thèse s'inscrivent dans les problématiques de modélisation d'environnement pour la localisation par vision monoculaire. Nous nous intéressons tout particulièrement à la modélisation des environnements intérieurs dynamiques. Les environnements intérieurs sont constitués d'une multitude d'objets susceptibles d'être déplacés. Ces déplacements modifient de façon notable la structure et l'apparence de l'environnement et perturbent les méthodes actuelles de localisation par vision. Nous présentons dans ces travaux une nouvelle approche pour la modélisation d'un environnement et son évolution au fil du temps. Dans cette approche, nous définissons la scène explicitement comme une structure statique et un ensemble d'objets dynamiques. L'objet est défini comme une entité rigide qu'un utilisateur peut prendre et déplacer et qui est repérable visuellement.

Nous présentons tout d'abord comment détecter et apprendre automatiquement les objets d'un environnement dynamique. Alors que les méthodes actuelles de localisation filtrent les incohérences dues aux modifications de la scène, nous souhaitons analyser ces modifications pour extraire des informations supplémentaires. Sans aucune connaissance *a priori*, un objet est défini comme une structure rigide ayant un mouvement cohérent par rapport à la structure statique de la scène. En associant deux méthodes de localisation par vision reposant sur des paradigmes différents, nous comparons les multiples passages d'une caméra dans un même environnement. La comparaison permet de détecter des objets ayant bougé entre deux passages. Nous pouvons alors, pour chaque objet détecté, apprendre un modèle géométrique et un modèle d'apparence et retenir les positions occupées par l'objet dans les différentes explorations. D'autre part, à chaque nouveau passage, la connaissance de l'environnement est enrichie en mettant à jour les cartes métrique et topologique de la structure statique de la scène.

La découverte d'objet par le mouvement repose en grande partie sur un nouvel algorithme de détection de multiples structures entre deux vues que nous proposons dans ces travaux. Etant donné un ensemble de correspondances entre deux vues similaires, l'algorithme, reposant sur le RANSAC, segmente les structures correspondant aux différentes paramétrisations d'un modèle mathématique. La méthode est appliquée à la détection de multiples homographies pour détecter les plans de la scène et à la détection de multiples matrices fondamentales pour détecter les objets rigides en mouvement.

La modélisation de l'environnement que nous proposons est utilisée dans une

nouvelle formulation de reconnaissance de lieu prenant en compte la connaissance d'objets dynamiques susceptibles d'être présents dans l'environnement. Le modèle du lieu est constitué de l'apparence de la structure statique observée dans ce lieu. Une base de données d'objets est apprise à partir des précédentes observations de l'environnement avec la méthode de découverte par le mouvement. La méthode proposée permet à la fois de détecter les objets mobiles présents dans le lieu et de rejeter les erreurs de détection dues à la présence de ces objets.

L'ensemble des approches proposées sont évaluées sur des données synthétiques et réelles. Des résultats qualitatifs et quantitatifs sont présentés tout au long du mémoire.

Abstract

In this thesis, we explore the problem of modeling an unknown environment using monocular vision for localization applications. We focus in modeling dynamic indoor environments. Many objects in indoor environments are likely to be moved. These movements significantly affect the structure and appearance of the environment and disrupt the existing methods of visual localization. We present in this work a new approach for modeling the environment and its evolution with time. We define explicitly the scene as a static structure and a set of dynamic objects. The object is defined as a rigid entity that a user can take, move and that is visually detectable.

First, we show how to automatically discover new objects in a dynamic environment. Existing methods of visual localization simply ignore the inconsistencies due to changes in the scene. We aim to analyze these changes to extract additional information. Without any prior knowledge, an object is a set of points with coherent motion relative to the static structure of the scene. We combine two methods of visual localization to compare various explorations in the same environment taken at different time. The comparison enables to detect objects that have moved between the two shots. For each object, a geometric model and an appearance model are learned. Moreover, we extend the scene model while updating the metrical map and the topological map of the static structure of the environment.

Object discovery using motion is based on a new algorithm of multiple structures detection in an image pair. Given a set of correspondences between two views, the method based on RANSAC extracts the different structures corresponding to different model parameterizations seen in the data. The method is applied to homography estimation to detect planar structures and to fundamental matrix estimation to detect structures that have been shifted one from another.

Our approach for dynamic scene modeling is applied in a new formulation of place recognition to take into account the presence of dynamic objects in the environment. The model of the place consists in an appearance model of the static structure observed in that place. An object database is learned from previous observations in the environment with the method of object discovery using motion. The place recognition we propose detects the dynamic objects seen in the place and rejects the false detection due to these objects.

The different methods described in this dissertation are tested on synthetic and real data. Qualitative and quantitative results are presented throughout the disser-

tation.

Introduction

Le monde dans lequel nous évoluons a une structure tri-dimensionnelle dont l'agencement et l'apparence sont susceptibles d'être modifiés dans le temps. Nous sommes capables de nous déplacer dans ce monde, de le décrire et de reconnaître des objets ou des personnes car notre cerveau interprète l'ensemble des informations captées par la rétine de nos yeux en utilisant des connaissances propres. Si nous pouvons reproduire le sens de la vue gràce aux caméras numériques, les systèmes de vision par ordinateur s'attachent à reproduire les capacités d'analyse que l'homme utilise généralement pour interpréter les images qu'il perçoit.

La vision par ordinateur est une branche de l'intelligence artificielle qui a pour vocation de fournir des systèmes informatiques capables d'extraire de l'information à partir d'un ensemble d'images ou d'un flux vidéo. Les applications sont multiples : la construction de modèles numériques 3D, la reconnaissance de formes ou d'objets, la détection d'évènements, la recherche d'information dans des bases d'images en sont quelques-unes. Parallèlement aux avancées de la recherche en vision par ordinateur, l'essor des outils de capture d'images numériques associés à des optiques de qualité a rendu possible le développement de nouvelles technologies issues de ce domaine. Ainsi, les appareils photos numériques grand puplic intègrent déjà des systèmes de détection automatique de visage, de suivi de visage ou de création de panorama.

On compte parmi les applications importantes de la vision par ordinateur, la localisation par vision et l'aide à la navigation. Ces applications nécessitent de modéliser les environnements que l'on veut explorer à partir d'images ou de séquences vidéo. Aujourd'hui, de nombreuses sources d'images numériques existent. Des images de villes du monde entier sont disponibles avec *Google Street View* et sont déjà utilisées dans un logiciel d'aide à la navigation (*Google Navigation*). Il s'agit en grande partie d'images extérieures, mais on peut aussi visiter l'intérieur de quelques magasins de grande ville en profitant d'images panoramiques. D'autre part, avec la démocratisation des caméras numériques et des ordiphones et l'apparition de nouveaux modes de communication, le partage de photos et de vidéos est de plus en plus populaire. Toutes ces données mises en ligne peuvent être utilisées pour modéliser les environnements observés et fournir une description exploitable pour se localiser et se déplacer.

Une approche qui s'est fortement développée est l'extraction d'informations tridimensionnelles à partir d'images. Il est possible à partir de deux images d'une même scène, prises de points de vue différents, de déduire des informations 3D et notam-

ment des valeurs de profondeur. La combinaison des informations extraites dans plusieurs images permet de reconstruire un modèle numérique 3D de la scène (figure 1). Ce modèle peut être ensuite utilisé pour localiser les caméras correspondant à chaque prise de vue.

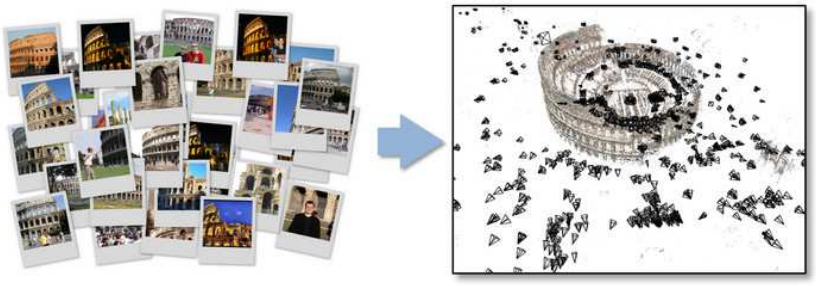


FIGURE 1 – Projet *Rome in a day* : Reconstruction 3D du Colisée à partir d'images Flickr

Ces dernières années, les communautés scientifiques de robotique et de vision par ordinateur ont travaillé à l'élaboration d'un système autonome de localisation temps réel à partir du flux vidéo provenant d'une ou plusieurs caméras. Le système doit calculer en temps réel les positions 3D des caméras et conjointement reconstruire l'environnement en 3D.

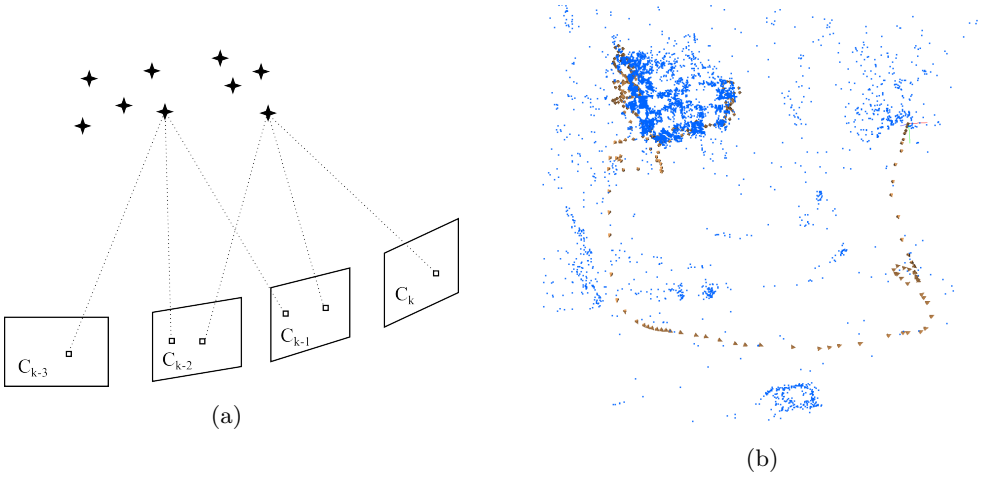
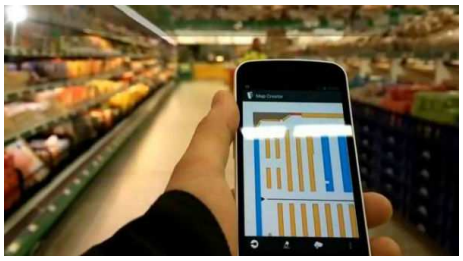


FIGURE 2 – Reconstruction 3D et localisation. (a) Observation d'un point 3D dans plusieurs images prises de points de vue différents. (b) Modèle numérique 3D de la scène et trajectoire de la caméra reconstruits

L'application directe de cette approche est la localisation en milieu intérieur ou *géolocalisation indoor*. Il existe aujourd'hui des systèmes de géolocalisation indoor pour smartphones reposant sur le traitement de données non visuelles (figure 3). L'objectif des études menées en vision par ordinateur est de fournir un système de positionnement équivalent au GPS en utilisant uniquement des données visuelles. L'avantage est d'utiliser des dispositifs simple à mettre en œuvre (caméras numériques). On peut aussi imaginer qu'une banque d'images soient déjà disponible sur



(a)

(b)

FIGURE 3 – Géolocalisation Indoor : (a) Système IndoorAtlas reposant sur l'analyse de champs magnétiques terrestre, (b) Système Insiteo reposant sur l'analyse de signaux WI-FI et GPS.

internet et qu'elle soit mise à jour régulièrement. Un tel système permettrait des applications d'aide à la navigation dans des grands environnements fréquemment fréquentés comme les musées, les aeroports ou les centres commerciaux.

La reconstruction 3D ouvre aussi la voie à des applications de réalité augmentée. La réalité augmentée (RA) offre la possibilité de plonger le spectateur dans un monde partiellement réel et permet une intéraction avec les objets virtuels. Les systèmes

(a)

(b)

(c)

FIGURE 4 – Quelques exemples d'applications de réalité augmentée

RA incrustent de façon réaliste des objets virtuels dans la séquence vidéo en temps réel. Les applications sont multiples et touchent de plus en plus de domaines, tels que les jeux vidéo, le cinéma et la télévision, les industries (aide à la maintenance ou à l'assemblage). Dans le domaine du tourisme, la réalité augmentée peut être utilisée pour recréer virtuellement un patrimoine disparu. Elle a été ainsi mise en œuvre

à l'abbaye de Cluny pour une visite virtuelle (figure 4 (a)). La réalité augmentée devient aussi une nouvelle stratégie de communication et intervient dans la publicité, la promotion d'albums musicaux ou de romans (figure 4 (b) et (c)).

Les travaux présentés dans ce mémoire ont pour vocation de contribuer à la problématique de la modélisation des environnements à partir de données visuelles pour des applications de localisation. Nous nous intéressons particulièrement à la modélisation des environnements dynamiques. En particulier, nos travaux visent à étudier comment exploiter les differentes explorations dans un même environnement prises à des instants différents pour modéliser au mieux le milieu et fournir une description exploitable dans des applications de localisation. Les bureaux d'une entreprise, les ateliers d'une usine ou les rayons de supermarché sont des exemples d'environnements qui peuvent être difficiles à modéliser : ce sont des environnements de grande taille et dynamiques. Dans ce type de milieu, le modèle construit à un instant précis devient rapidement obsolète. La modélisation doit pouvoir prendre en compte les changements survenus dans l'environnement et séparer les parties statiques et dynamiques pour fournir une carte cohérente.

Les approches métriques et topologiques de la localisation et cartographie par vision

Ces dix dernières années, de nombreux travaux ont été consacrés à la résolution du problème de localisation et cartographie simultanées par vision (*Simultaneous localization and Mapping* ou SLAM). Il s'agit, à partir d'images acquises par un périphérique de capture explorant un environnement *a priori* inconnu ou partiellement connu, d'estimer la position du capteur tout en cartographiant l'environnement au fur et à mesure de l'exploration. On parle de système par vision monoculaire lorsque le capteur utilisé est une caméra à champ moyen. D'autres systèmes utilisent des caméras stéréoscopiques ou panoramiques ou des ensembles de caméras rigidement liés. Quel que soit le type de caméra utilisé, les algorithmes reposent uniquement sur l'information extraite des images. Ces derniers peuvent être classés dans deux catégories.

Les approches métriques construisent une carte éparse des points 3D de l'environnement et estiment les coordonnées précises de la caméra dans l'espace. On peut, à partir de deux images (ou plus) d'un même lieu prises de points de vue différents, obtenir des informations tridimensionnelles. Le traitement des images successives d'un flux vidéo permet de reconstruire incrémentalement un modèle 3D précis de la scène. Ces approches offrent de nombreuses possibilités d'applications RA. Elles sont cependant sujettes à une dérive dans l'estimation de la caméra mobile et par conséquent dans l'estimation de la carte entière et ne sont pas adaptées aux environnements de grande taille.

Dans les approches basées sur l'apparence ou approches topologiques, l'environnement est segmenté en lieux distincts qui forment les noeuds d'un graphe ou *carte topologique*. Les arêtes du graphe encodent alors les relations entre les lieux. Ces

approches reposent sur la reconnaissance de lieu. Elles contournent de cette manière les problèmes d'accumulation d'erreur des systèmes de SLAM métrique. Elles sont, par conséquent, plus adaptées aux environnements de grande taille et permettent de détecter précisément les cycles dans la trajectoire de la caméra.

La problématique des environnements dynamiques

Les premières solutions au problème du SLAM sont souvent associés aux travaux décrits dans [Smith et Cheeseman, 1986,Durrant-Whyte, 1987,Moutarlier et Chatila, 1989]. Ces approches considèrent un environnement statique. Les éléments modélisés dans la carte sont fixes. De cette façon, les positions des éléments peuvent être déterminées de manière incrémentale et servir ensuite de points de repère pour localiser le capteur dans la carte. Cependant, les milieux réels, et notamment les milieux intérieurs, sont souvent composés d'une quantité d'objets susceptibles d'être déplacés. Ces déplacements d'objets modifient de manière notable l'apparence et la structure d'une scène. La carte apprise à un certain instant devient alors obsolète et la localisation dans l'environnement est perturbée. Une solution qui présuppose que le milieu exploré est statique fournira une carte incohérente. Un système résolvant le problème du SLAM ne doit pas considérer un environnement statique mais prendre en compte de manière explicite la présence potentielle d'objets dynamiques. Il peut les détecter ou les ignorer mais ne peut pas les utiliser pour mettre à jour la carte de l'environnement en les supposant statiques.

D'autre part, les mouvements dans la scène peuvent aussi être une source d'information supplémentaire non négligeable. Les objets déplacés sont susceptibles d'être observés plusieurs fois dans l'environnement, ils font donc partie intégrante de la scène. A ce titre, l'apparence et la structure de chaque objet dynamique doivent aussi être enregistrées dans le modèle de l'environnement. Les objets dynamiques deviennent une source d'information exploitable par un système autonome lors d'une exploration ultérieure.

Dans ces travaux, nous proposons une nouvelle approche pour modéliser un environnement en tenant compte de la présence d'objets dynamiques. Le but est de façonner, à partir des données visuelles provenant d'une ou plusieurs explorations d'une caméra monoculaire mobile dans un milieu *a priori* inconnu, la description la plus juste de l'environnement à un instant t. Cette approche fait intervenir la modélisation de la structure statique, la localisation de la caméra dans l'environnement et la modélisation des objets dynamiques. Nous détaillons cette approche ainsi que les contributions de ces travaux de thèse dans la section suivante.

Approche proposée et contributions

Nous nous situons dans le contexte de la modélisation de milieux intérieurs dynamiques. Ces milieux sont constitués d'une multitude d'objets susceptibles de bouger.

S'ils sont souvent déplacés, ces objets restent de manière permanente dans l'environnement et il est nécessaire de les prendre en compte dans la modélisation de la scène.

Nous proposons dans ces travaux une nouvelle formulation de la modélisation d'un environnement à partir des données visuelles provenant d'une caméra monoculaire. L'environnement est constitué de l'ensemble des lieux explorés par un ou plusieurs utilisateurs. Nous le définissons explicitement comme une structure statique d'une part et une quantité d'objets dynamiques d'autre part. La structure statique est par définition fixe et immuable mais la modélisation de la structure statique est susceptible d'évoluer au fil du temps grâce aux informations extraites des diverses explorations dans l'environnement. L'approche proposée exploite ces multiples explorations pour tirer le maximum d'informations sur la scène et son évolution au fil du temps :

- L'agencement et l'apparence de la structure statique sont mis à jour à chaque nouvelle exploration, ce qui permet d'améliorer la localisation d'une caméra mobile dans une carte actuelle de l'environnement.
- De nouveaux objets sont automatiquement détectés et appris grâce la comparaison des explorations dans l'environnement.

L'objectif est de produire une description valable et cohérente de l'environnement. Cette description est ensuite exploitable dans des applications de localisation.

L'approche que nous proposons fait intervenir la notion d'objet. Nous définissons un objet comme une entité rigide qu'un utilisateur peut prendre et déplacer et que l'on peut repérer. Dans le contexte de découverte de nouveaux objets par vision, l'objet, tel que nous le définissons, possède les deux propriétés suivantes :

- il garde une cohérence dans le mouvement,
- il est décelable visuellement.

Un objet est donc une structure ayant un mouvement cohérent par rapport à la structure statique de l'environnement. Cette définition implique une notion de persistance de l'objet qui doit être observé plusieurs fois, dans le même lieu ou dans des lieux différents.

L'approche que nous proposons est illustrée figure 5. Dans notre contexte, la modélisation de la scène fait intervenir de multiples utilisateurs se déplaçant, caméra à la main, dans le même milieu. Nous proposons un outil de collaboration permettant de récolter les informations des diverses explorations. Chaque exploration permet de construire un modèle de la scène. La comparaison de deux explorations fournit des informations supplémentaires, notamment sur les changements survenus entre les deux passages et permet ainsi d'inférer la présence d'objet. La représentation de la scène est enrichie à partir de cette comparaison : la structure statique est mise à jour et les structures dynamiques détectées sont enregistrées dans une base de données.

Nos contributions s'articulent autour de deux axes de recherche que nous avons suivis pendant ces travaux de thèse ; à savoir la localisation par vision en milieu inconnu ou partiellement connu et la découverte et l'apprentissage de nouveaux objets. Les contributions majeures sont détaillées dans les sous-sections suivantes.

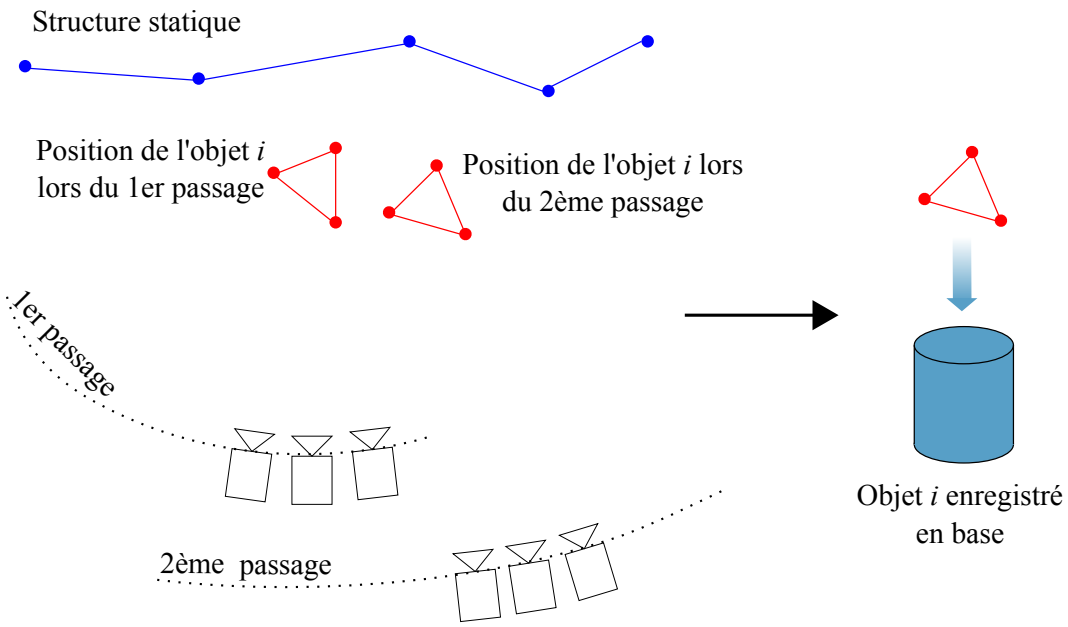


FIGURE 5 – **Approche proposée.** La comparation de deux explorations et la détection des changements survenus permet d'enregistrer les modèles de la structure statique et des structures dynamiques

Relocalisation d'une caméra dans une carte métrique 3D

L'association d'une méthode de reconnaissance de lieu et d'une méthode métrique répondant au problème du SLAM est utilisée pour se relocaliser dans une carte 3D et détecter des incohérences entre deux explorations dans une même scène faites à des instants différents. Ces travaux ont été publié dans [Decrouez *et al.*, 2012b].

Extraction d'objet par le mouvement

Nous définissons un objet par son mouvement : c'est un ensemble de points ayant un mouvement cohérent par rapport à la structure statique de la scène. De nouveaux objets sont extraits en comparant deux séquences vidéos prises dans un même environnement. Pour chaque objet, nous apprenons un modèle d'apparence et un modèle 3D. Nous avons publié cette méthode dans [Decrouez *et al.*, 2012a]. De cette façon, l'association de méthodes de SLAM métrique et de SLAM topologique permet de détecter et d'apprendre de nouveaux objets et d'enrichir la connaissance de la scène explorée.

Detection de multiples structures entre deux vues

Dans le but d'expliquer les incohérences entre deux explorations différentes et de découvrir de nouveaux objets, nous avons développé un nouvel algorithme permettant de détecter les différentes structures présentes dans une scène à partir d'une

paire d'images similaires. Nous avons publié ces travaux dans [Decrouez *et al.*, 2012c]. L'algorithme est appliqué à la détection de multiples homographies et à la détection de multiples matrices fondamentales. Il repose en partie sur la méthode d'échantillonnage BetaSAC que nous avons publié dans [Méler *et al.*, 2010]. BetaSAC permet d'accélerer l'algorithme RANSAC qui estime les paramètres d'un modèle mathématique à partir d'un ensemble de données bruitées. Les travaux effectués en début de thèse ont permis d'évaluer les performances de BetaSAC et de se comparer à l'état de l'art.

Nouvelle méthode de reconnaissance de lieu

Nous avons formulé une nouvelle approche de reconnaissance de lieu dans un environnement partiellement connu : les objets dynamiques susceptibles d'être présents dans la scène sont préalablement appris et leurs apparences sont enregistrées dans une base de données. Nous avons publié ces travaux dans [Decrouez *et al.*, 2011]. Chaque lieu est modélisé par une partie statique et un ensemble d'objets potentiellement présents. A chaque nouvelle image, la vraisemblance de l'observation est calculée en supposant la présence d'un ou plusieurs objets. La formulation que nous proposons permet de rejeter une grande partie des fausses détections dues au mouvement d'objets dynamiques.

Organisation du manuscrit

L'étude rapportée dans ce manuscrit porte sur le problème de la modélisation de la structure et de l'apparence d'un environnement *a priori* inconnu à partir d'un ensemble de séquences vidéos acquises avec une caméra monoculaire à champ moyen. Le manuscrit est organisé en cinq chapitres que nous résumons ici.

Le chapitre 1 présente en quatre sections la problématique de la modélisation d'un environnement *a priori* inconnu. Nous présentons dans un premier temps le problème de la localisation et cartographie simultanées (ou SLAM pour *Simultaneous Localization and Mapping*), problème qui a été posé en 1986 lors de la conférence internationale de robotique et d'automatique (ICRA) [Durrant-Whyte et Bailey, 2006] et qui, depuis, intéresse vivement différentes communautés scientifiques notamment dans les domaines de la robotique et de la vision par ordinateur. Ce chapitre a aussi pour but de donner une vue d'ensemble des approches proposant un système de localisation d'une caméra mobile dans un environnement inconnu ou partiellement connu. Nous présentons, en particulier, les méthodes métriques qui reposent sur la reconstruction éparse de l'environnement en 3D et l'utilisation de la carte métrique obtenue pour la localisation de la caméra. Nous examinons également les approches *topologiques* qui décrivent l'environnement sous la forme d'un graphe dont les noeuds correspondent à des lieux distincts et dont les liens peuvent servir à encoder différents types d'information telle que la similarité, l'adjacence spatiale. De la même manière, la carte obtenue est utilisée pour la localisation symbolique d'une caméra mobile. Dans la troisième section, nous exposons la problématique de la modélisation des en-

vironnements dynamiques. Notamment, nous nous intéressons aux environnements intérieurs composés d'une multitude d'objets susceptible de bouger, objets qui font partie intégrante de l'environnement et qui viennent perturber les méthodes présentées. Nous analysons ensuite les approches utilisant la notion d'objet pour améliorer la localisation et affiner la modélisation de l'environnement.

Dans le chapitre 2, nous présentons en cinq sections les principes et les algorithmes utilisés dans ce mémoire. Nous présentons deux méthodes de localisation dans un environnement *a priori* inconnu. Le système de localisation et cartographie par vision développé par [Mouragnon *et al.*, 2006] est décrit dans la première section. Nous expliquons la chaîne de traitements appliqués au flux vidéo fourni par une caméra numérique à champ moyen. Cette méthode métrique repose sur la représentation de l'image par un ensemble de primitives locales, l'association des primitives dans les images successives pour reconstruire l'environnement en 3D et la trajectoire de la caméra en temps réel et l'optimisation de la carte par une méthode d'ajustement de faisceaux local. Nous présentons ensuite les limitations de cette méthode avec le problème de dérive sur les longues trajectoires et le comportement en milieux dynamiques. Ce-dernier est illustré par une évaluation qualitative de l'algorithme sur une séquence de synthèse comportant des objets en mouvement. Nous présentons dans la troisième partie une méthode de reconnaissance de lieu reposant sur le critère de maximum de vraisemblance qui a l'avantage d'être simple à mettre en œuvre. Les différentes hypothèses de lieu sont mises en correspondance avec l'observation courante avec le calcul d'un score de similarité entre les histogrammes de mots visuels associés aux images. Nous détaillons aussi les méthodes de validation géométrique permettant de rejeter les fausses détections. Nous décrivons le comportement de cette méthode en milieux dynamiques que nous illustrons par des résultats obtenus sur des données réelles. Ces deux méthodes fournissent les données d'entrée de notre étude et de l'algorithme d'extraction d'objet présenté dans le chapitre 3. Nous présentons ensuite la méthode d'optimisation robuste RANSAC qui permet d'estimer les paramètres d'un modèle à partir d'un ensemble de données bruitées contenant des valeurs aberrantes. La stratégie d'échantillonage BetaSAC est détaillée dans cette partie : cette nouvelle méthode permet d'accélérer considérablement le RANSAC en prenant en compte des *a priori* connus. Enfin, nous détaillons dans la dernière partie l'algorithme *Mean-Shift* qui est une approche non-paramétrique permettant de détecter les modes d'une distribution de probabilité. Cette méthode est utilisée dans le nouvel algorithme de détection de multiples structures entre deux vues décrit dans le chapitre 4.

Le chapitre 3 est dédié à la présentation d'une nouvelle méthode d'extraction automatique d'objets. La méthode met en œuvre la comparaison de deux séquences vidéos prises dans une même scène à des instants différents. Les changements survenus entre les deux passages permettent d'inférer la présence d'objet : les objets sont extraits par le mouvement. Les concepts développés dans ce chapitre sont exposés dans la première partie, notamment la définition de l'objet et la définition de la scène. La deuxième partie résume un état de l'art des méthodes d'extraction d'objet par l'apparence et par le mouvement. Nous présentons ensuite notre algorithme d'extraction automatique d'objet qui utilise uniquement les images provenant d'une caméra monoculaire. La combinaison d'une méthode métrique répondant au problème du SLAM et d'une méthode de reconnaissance de lieu permet de relocaliser la

caméra mobile dans les cartes métrique et topologique de l'environnement. L'analyse de la relocalisation est traitée dans la partie suivante : la détection d'incohérence dans les différentes cartes permet de déduire la présence éventuelle d'un ou plusieurs objets. Nous montrons comment valider l'hypothèse avec un algorithme de détection de multiples structures. Nous exposons, par la suite, les difficultés rencontrées pour déterminer et mettre à jour la structure statique de l'environnement de façon précise, notamment lorsqu'une grande partie de la scène a changé, ainsi que la solution retenue. Enfin, nous évaluons l'algorithme complet sur trois jeux de données réelles, constitués de séquences prises dans des environnements intérieurs. La première expérience met en œuvre la modélisation d'un milieu intérieur de grande taille. Plusieurs objets sont déplacés dans la scène en restant dans le même lieu. Nous explorons dans une deuxième expérience deux bureaux d'une entreprise. Des objets ont été bougés au sein d'un même lieu, ou sont déplacés dans un lieu différent. La troisième expérience est constituée de l'observation d'une maquette articulée de voiture et présente une autre application de l'algorithme qui permet de détecter le déplacement des sous-parties rigides de la maquette les unes par rapport aux autres. Ces trois expériences attestent les performances de l'approche proposée pour la découverte de nouveaux objets.

Dans le chapitre 4, nous étudions le problème de détection de multiples structures et nous présentons une nouvelle solution à ce problème. La détection des différents plans ou des différents mouvements dans une scène 3D est une étape essentielle à la méthode d'extraction automatique d'objet présentée dans le chapitre 3. Nous présentons dans une première partie un état de l'art des méthodes existantes et les limites de ces méthodes. Nous présentons ensuite une vue d'ensemble de l'algorithme puis les différentes étapes sont détaillées dans les parties suivantes. L'algorithme proposé repose sur le RANSAC qui est une procédure itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données. Nous adaptons l'algorithme RANSAC pour faire correspondre plusieurs modèles au jeu de données. Tout d'abord, deux nouvelles stratégies d'échantillonnage utilisées pour générer de nouvelles hypothèses sont présentées puis évaluées sur des données synthétiques et réelles. Ces stratégies sont basées sur l'algorithme BetaSAC. La première stratégie prend en compte des informations de proximité spatiale dans l'échantillonage. La seconde utilise les informations provenant des précédentes hypothèses. Les résultats prouvent l'efficacité des stratégies dans le contexte de détection de multiples structures. Nous présentons ensuite la méthode d'optimisation locale utilisée pour accélérer le processus. Cette méthode permet d'obtenir une paramétrisation plus précise à chaque fois qu'une nouvelle hypothèse est générée. Enfin nous présentons la méthode de fusion des hypothèses permettant de garder les différents modèles expliquant au mieux les données. Cette étape met en œuvre l'évaluation de chaque hypothèse générée par l'étude de l'erreur résiduelle des points. L'algorithme complet est évalué sur les problèmes d'estimation d'homographie et de matrice fondamentale à partir des correspondances de points entre deux vues. Les données de test sont des données synthétiques et des données réelles bruitées. Les résultats montrent l'efficacité de l'algorithme sur la détection de multiples homographies et la détection de multiples matrices fondamentales.

Enfin, nous présentons dans le chapitre 5 une application de la modélisation que nous proposons. Un nouvel algorithme de reconnaissance de lieu tenant compte

de la connaissance d'objets susceptibles d'être présents dans la scène est exposé. Nous rappelons tout d'abord succinctement le problème du SLAM dans sa forme probabiliste qui consiste à calculer, à chaque instant t et pour chaque lieu, la probabilité de se trouver dans le lieu. Connaissant l'ensemble des observations jusqu'à t et la position précédente, la probabilité peut être estimée par filtrage bayésien. Le lieu est déterminé au sens du maximum *a posteriori*. Le filtrage bayésien fait intervenir le calcul de vraisemblance de l'observation qui est la probabilité de faire une observation à l'instant t étant donné le modèle de l'environnement à $t-1$. Ce calcul de vraisemblance correspond globalement à une mesure de similarité entre l'image courante et les lieux lieux déjà enregistrés. Nous proposons de modifier le calcul de vraisemblance en tenant compte de l'existence d'objets dynamiques ; l'apparence de ces objets étant apprise par la méthode d'extraction par le mouvement sur des explorations précédentes. Pour chaque hypothèse de lieu, nous supposons la présence d'un ou plusieurs objets dynamiques en combinant le modèle d'apparence associé au lieu en question et ceux associés aux objets supposés présents. Le calcul de vraisemblance que nous proposons peut intervenir soit dans un algorithme de reconnaissance de lieu s'appuyant sur le critère de maximum de vraisemblance, soit dans le calcul de probabilité de se trouver dans un lieu dans le cadre d'une méthode probabiliste. Nous évaluons cette nouvelle formulation sur une séquence réelle. Au cours de l'expérience, on détermine le lieu et les objets correspondant au maximum du score de similarité donné par le calcul de vraisemblance. La méthode permet de rejeter les fausses détections dues à la présence d'objets dynamiques.

Nous détaillons dans l'annexe A les concepts de géométrie projective avec le modèle de caméra perspective et la géométrie qui leur est associée.

Chapitre 1

Modélisation d'un environnement inconnu

Contenu du chapitre

Ce premier chapitre a pour but de donner une vue d'ensemble des approches existantes pour la localisation d'une unique caméra sans connaissance a priori sur l'environnement parcouru. Nous présentons aussi les approches utilisant la notion d'objet pour aider et améliorer la localisation.

1.1 Le problème de la localisation et la cartographie simultanées

Le problème de la cartographie et localisation simultanées ou SLAM *(Simultaneous Localization and Mapping)* consiste, pour un robot placé à un endroit inconnu dans un environnement inconnu, à construire une carte précise de l'environnement et à se localiser, à partir des données qu'il acquiert au fur et à mesure de son déplacement. Ce problème a été soulevé par la communauté robotique en 1986 lors de la conférence internationale de robotique et d'automatique (ICRA) [Durrant-Whyte et Bailey, 2006] : une solution pouvant fournir la position d'un robot sans aucune information *a priori* permettrait de développer des systèmes de navigation complètement autonomes à la géolocalisation. Si on considère la *localisation* et la *cartographie* séparément, chaque problème semble simple à résoudre. L'interprétation des informations fournies par un ou plusieurs capteurs peut permettre soit de se localiser dans une carte déjà disponible, soit de construire un modèle de l'environnement si on connait les positions de ces capteurs. Un système résolvant le problème du SLAM doit effectuer les deux tâches simultanément : il s'agit de déterminer la position courante du robot ou du capteur dans la carte construite sur la base d'une observation de l'environnement jusqu'à un certain instant, puis de mettre à jour la carte avec les nouvelles informations obtenues à partir de cette position. Les deux tâches dépendent fortement l'une de l'autre. Les traiter de front sans aucune connaissance *a priori* est particulièrement difficile : la trajectoire du robot et la carte de l'environnement sont inconnues et leurs estimations sont corrélées. Les premiers travaux sur la localisation en environnement inconnu sont souvent associés aux travaux décrits dans [Smith et Cheeseman, 1986, Durrant-Whyte, 1987, Moutarlier et Chatila, 1989]. Dans ces articles, des outils statistiques sont définis pour représenter les positions de points de repère de l'environnement et celles du robot et de manipuler les incertitudes des positions de tous ces objets avec un filtre de Kalman étendu. Ces premiers algorithmes considèrent uniquement les environnements statiques. Ils montrent qu'il existe une forte corrélation entre les positions estimées du robot et des amers extraits de l'environnement observé et que cette corrélation s'accentue au fur et à mesure des observations. Ces avancées théoriques ont ouvert la voie à la conception de systèmes de localisation et cartographie simultanées que nous présentons ci-après. Nous tentons dans les sections suivantes de dresser une nomenclature des approches en fonction du type de capteur utilisé et du type de carte produite par l'algorithme. Nous nous attachons ensuite à décrire les approches reposant uniquement sur la vision.

Dans ce chapitre et dans la suite du mémoire, nous parlons de « méthodes de SLAM ». Il s'agit plus précisément de méthodes répondant au problème du SLAM.

1.1.1 Les différents types de capteurs

Différents types de capteurs ont été mis en œuvre pour examiner la problématique du SLAM. Les capteurs de distance tels que télémètres laser, radars ou sonars ont été largement utilisés. Le système proposé dans [Crowley, 1989] modélise l'environnement parcouru par un robot mobile équipé d'un capteur à ultrasons. L'auteur utilise une carte de l'environnement apprise au préalable qui figure une description partielle de l'environnement et maintient cette carte mise à jour en temps réel en manipulant les estimations des positions avec un filtre de Kalman. Ces premiers travaux considèrent un milieu statique. Les auteurs de [Crowley *et al.*, 1989] proposent une extension de ces travaux adaptée aux milieux dynamiques. Le système détecte les segments stationnaires qui sont cohérents avec la carte apprise hors ligne et restreint la correction de la position aux segments stationnaires.

A partir des années 2000, parallèlement à l'essor des caméras et appareils photos numériques, de nombreux travaux se sont attachés à résoudre le problème du SLAM en utilisant des capteurs d'images numériques [Davison, 2003, Eade et Drummond, 2006, Mouragnon *et al.*, 2006, Davison *et al.*, 2007]. On parle de SLAM visuel ou de SLAM par vision. Une caméra représente en effet un dispositif simple à mettre en œuvre. D'autre part, l'information encodée dans une image est plus riche que celle donnée par les capteurs de distance, notamment en terme d'apparence. On peut néanmoins noter que l'extraction et la mise en correspondance des primitives visuelles sont plus complexes et exigent un temps de calcul plus important.

1.1.2 Les différents types de cartes

Les méthodes de SLAM, et notamment les méthodes de SLAM par vision, peuvent aussi être classées en fonction du type de carte qu'elles manipulent. On distingue alors les approches métriques qui s'intéressent aux propriétés géométriques de l'environnement et les approches topologiques qui s'attachent à décrire la connectivité des différents lieux.

Les premières représentations utilisées dans des approches métriques sont les grilles d'occupation [Elfes, 1990, Moravec, 1988]. Ces solutions modélisent l'environnement en deux dimensions par une grille dont chaque case encode le caractère occupé ou libre d'une portion de l'espace grâce à une valeur binaire.

Une autre solution est la représentation de l'environnement sous la forme d'un ensemble d'amers de positions géométriques connues. Un amer est un point 3D repérable dans l'environnement. La position du capteur est déterminée à partir des positions des amers reconnus. Les auteurs de [Chenavier et Crowley, 1992] localisent un robot mobile muni d'une caméra dans un environnement connu : les positions de certains points de repère ayant été appris au préalable. Le système utilise un filtre de Kalman étendu pour localiser le robot en fusionnant les données odométriques et les données visuelles.

Les approches topologiques utilisent une représentation complètement différente :

l'environnement est modélisé sous la forme d'un graphe de lieux dont les arêtes représentent les relations entre les lieux. Les nœuds des cartes topologiques sont souvent déduits des informations perceptuelles provenant du capteur. Les auteurs de [Kuipers et Byun, 1991] proposent un sytème de navigation dans de grands environnements avec un robot muni de sonars. Au cours de son exploration, le système définit des lieux disctinctifs comme des points uniques correspondant aux extrema locaux de mesures de particularité, par exemple la mesure de « distances égales entre le robot et les objets voisins ». Les données provenant d'une caméra apportent une information riche sur l'apparence des lieux et sont adaptées à ce type d'approche.

Nous nous intéresserons, dans la section suivante, uniquement aux approches de SLAM par vision. Nous exposons les méthodes de localisation par vision que nous classons en deux catégories : les approches métriques et les approches topologiques. Nous présentons ensuite notre approche pour modéliser une scène dynamique uniquement à parir des données provenant d'une caméra monoculaire. Nous définissons alors la scène comme une structure statique d'une part et un ensemble d'objets dynamiques d'autre part.

1.2 Localisation et cartographie simultanées par vision monoculaire

Le problème de la localisation par vision monoculaire est le suivant. Une caméra est mobile dans un environnement *a priori* inconnu ou partiellement connu, sa première position est aussi inconnue. Il s'agit, en utilisant uniquement les images fournies par le flux vidéo, de modéliser l'environnement et de localiser la caméra dans l'environnement modélisé. Les approches existantes répondant au problème du SLAM peuvent être classées en deux catégories selon le modèle de représentation de l'environnement sous-jacent : les approches métriques et les approches topologiques. Les algorithmes de SLAM métrique se concentrent sur l'estimation précise de la trajectoire de la caméra et des positions 3D des points caractéristiques de l'environnement. Les approches topologiques reposent sur une représentation discrète et symbolique de l'environnement : on cherche à caractériser l'apparence d'un lieu et non la structure géométrique. L'environnement est représenté sous la forme d'un graphe. Les nœuds du graphe correspondent à des lieux distincts et les arêtes représentent les relations entre lieux (positionnement relatif, adjacence temporelle, score de similarité). Cette représentation est notamment plus adaptée aux environnements de grande taille et permet une navigation symbolique pour atteindre un lieu en particulier. Nous présentons dans les parties suivantes la représentation de l'image communément utilisée en vision par ordinateur puis un état de l'art sur le SLAM métrique et le SLAM topologique.

1.2.1 Représentation de l'image

En vision par ordinateur, l'image est généralement caractérisée par un ensemble de primitives visuelles. Une primitive visuelle constitue une information pertinente

de l'image correspondant généralement à un point remarquable par certaines caractéristiques (couleur, texture...). On parle aussi de point d'intérêt. La primitive est communément caractérisée par un descripteur. On peut alors comparer deux primitives entre calculant une distance entre leurs descripteurs respectifs. Le suivi de primitives dans plusieurs images successives permet de calculer les positions géométriques de points 3D de l'environnement ; c'est le principe du SLAM métrique. Dans une approche plus qualitative, la collection de primitives extraites dans une image peut être utilisée pour modéliser l'apparence du lieu associé à cette image.

1.2.2 Approches de SLAM métrique

Les approches de SLAM métrique reposent sur l'idée que l'observation d'un même point 3D de l'environnement à des instants différents permet de connaître sa position 3D par triangulation. Ainsi, la carte des points 3D de l'environnement est reconstruite au fur et à mesure des observations et il est possible de s'y localiser. Pour résoudre ce problème, des travaux en robotique s'appuient sur l'utilisation d'outil statistique tels qu'un filtrage Bayésien [Davison, 2003] ou un filtre de Kalman étendu [Davison *et al.*, 2007]. Les filtres de Kalman étendus (EKF pour *Extended Kalman Filter*) sont une extension du filtre de Kalman [Kalman, 1960] s'appliquant aux systèmes non linéaires. Ils remplacent les fonctions non linéaires par leur version linéarisée, grâce à un développement de Taylor. Dans une approche de SLAM reposant sur le filtre de Kalman, le vecteur d'état est composé de la position courante de la caméra et des positions de l'ensemble des points 3D. La matrice de covariance associée permet de quantifier l'incertitude sur chacune des données du vecteur d'état. Cette approche présente une limite directement liée au temps nécessaire à son fonctionnement. L'étape de mise à jour du vecteur d'état a une complexité en N^2, où N est la taille du vecteur d'état. Avec des trajectoires longues, la taille de la carte devient importante et les traitements en temps réel ne sont plus possibles.

D'autres travaux s'appuient sur des outils d'optimisation, comme l'ajustement de faisceaux [Triggs *et al.*, 2000]. L'idée de départ est de réaliser une reconstruction 3D à partir d'une collection d'images : les positions des caméras et des amers sont estimées hors ligne. On parle de *Structure from motion*. L'ajustement de faisceaux est une méthode de vision par ordinateur qui optimise l'ensemble des paramètres de la scène en minimisant une erreur caractérisant l'adéquation entre le modèle reconstruit et les observations du modèle. C'est la méthode de reconstruction 3D la plus précise mais elle ne permet pas une exécution en temps réel. Des travaux récents utilisent cet outil en ligne et de façon incrémentale pour reconstruire l'environnement en 3D. En particulier, les travaux de [Mouragnon *et al.*, 2006] et de [Klein et Murray, 2007] mettent en avant que la précision obtenue grâce à l'optimisation par ajustement de faisceaux est meilleure que pour les méthodes de SLAM s'appuyant sur le filtre de Kalman. Le système proposé par [Mouragnon *et al.*, 2006] utilise une méthode d'ajustement de faisceaux local (seuls les paramètres des dernières caméras sont optimisés) et parvient à reconstruire l'environnement en temps réel. Les méthodes reposant sur des filtres statistiques et sur des méthodes d'optimisation ont été comparées par les auteurs de [Strasdat *et al.*, 2010] sur des données synthétiques et réelles. Les auteurs confirment que la méthode par ajustement de faisceaux est plus précise.

Les approches temps réels présentent encore certaines limitations. On observe une dérive cumulative dans l'estimation de la pose de la caméra et une dérive du facteur d'échelle : les reconstructions sont réalisées à un facteur d'échelle théoriquement constant sur la séquence entière mais on observe une dérive de ce facteur le long de la trajectoire. Il est alors difficile de repérer que la caméra repasse par un même endroit et de détecter les boucles dans sa trajectoire. Il faut noter que la méthode PTAM de Klein et Murray [Klein et Murray, 2007] se localise à chaque nouvelle image dans la carte des points 3D reconstruite. On peut donc de cette façon détecter les boucles dans la trajectoire de la caméra mais cette stratégie limite la taille de l'environnement reconstruit. L'approche de [Mouragnon *et al.*, 2006] permet de se localiser par rapport aux points vus dans les dernières images. Le système permet de reconstruire des trajectoires longues mais il n'est pas possible de se relocaliser dans la carte lorsque la position de la caméra est perdue ou trop éloignée de sa position réelle.

1.2.3 Approches de SLAM topologique

Motivées par la détection de boucle et la localisation dans de grandes scènes, de nouvelles approches de SLAM topologique ou approches basées sur l'apparence ont été proposées dans la littérature [Ho et Newman, 2007, Cummins et Newman, 2009, Angeli *et al.*, 2008]. Ces méthodes considèrent le problème du SLAM comme un problème de reconnaissance d'image : deux images similaires proviennent probablement du même endroit. De nombreux systèmes de localisation reposent sur une représentation de l'image en sacs de mots visuels. Ces méthodes inspirées des techniques de recherche d'information présentent l'image comme un ensemble de primitives visuelles, les mots étant définis dans un dictionnaire ou vocabulaire [Sivic et Zisserman, 2003, Nister et Stewenius, 2006]. Les travaux de Cummins et Newman [Cummins et Newman, 2009] définissent un formalisme probabiliste reposant sur l'approche en sac de mots visuels. L'environnement est un ensemble de lieux discrets dont l'apparence est modélisée par une distribution sur les mots du dictionnaire. L'algorithme offre une robustesse remarquable grâce à la prise en compte des probabilités de co-occurrences des mots visuels (calculées hors-ligne) dans l'estimation de la vraisemblance de l'observation. Néanmoins, le système, qui a prouvé son efficacité sur des paires d'images stéréo n'autorise pas de traitements temps-réel. Les différentes méthodes mettent aussi en avant qu'une étape de validation géométrique nécessitant la connaissance précise des positions des points dans l'image améliorent considérablement la robustesse des résultats et devient souvent indispensable [Angeli *et al.*, 2008].

De plus en plus de travaux combinent les deux approches pour gérer des trajectoires plus longues tout en maintenant une carte des points 3D nécessaire aux applications de réalité augmentée. Les auteurs de [Castle *et al.*, 2008] proposent un système gérant simultanément plusieurs cartes 3D et mettant en œuvre la relocalisation de l'image courante. Ils recherchent parmi les images précédentes l'image la plus proche en utilisant une description globale de l'image.

1.3 La modélisation des environnements dynamiques

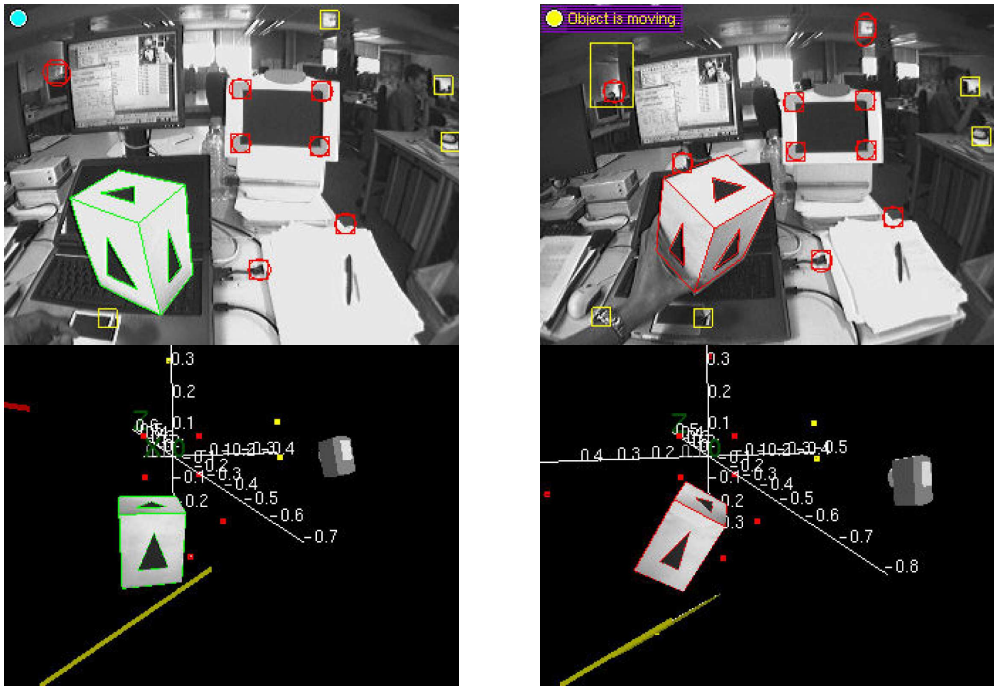
Les méthodes de SLAM métrique et topologique modélisent une scène statique constituée d'un ensemble de structures rigides et immobiles. Dans les approches métriques, les objets en mouvement sont traités de la même manière que les données aberrantes et sont filtrés par des méthodes robustes comme le RANSAC qui sera présenté dans le chapitre 2. Or, les environnements réels ne sont pas statiques. Ils sont constitués d'une multitude d'objets dynamiques ou d'objets temporaires qui paraissent statiques mais sont susceptibles d'être déplacés. Ces mouvements perturbent les méthodes de localisation précédemment citées. La solution consistant à filtrer les données aberrantes est possible dans des environnements faiblement dynamiques mais atteint rapidement ses limites lorsqu'une grande partie de la scène est modifiée. De plus, un objet qui parait statique mais qui peut être déplacé, est ajouté de manière définitive dans la carte de l'environnement. Ceci a pour conséquence des éventuelles incohérences dans la reconstruction de la trajectoire globale de la caméra.

De même, dans les approches topologiques reposant sur des algorithmes de reconnaissance d'image, les changements importants dus aux mouvements d'objets viennent contrarier la reconnaissance d'un lieu. Si un lieu n'est pas reconnu à cause du mouvement d'un objet, un nouveau lieu est ajouté à tort dans la carte topologique de l'environnement.

Ainsi, un système de SLAM se doit de prendre en compte les objets dynamiques pour modéliser un environnement réel : il ne doit pas ajouter un objet mobile à la carte et supposer qu'il reste statique car la carte fournie sera incohérente. Par ailleurs, les objets en mouvement font partie intégrante de la scène et fournissent une information aussi importante que les objets statiques. De nouvelles méthodes baptisées SLAMMOT (*Simultaneous Localisation and Mapping and Moving Object Tracking*) ont pour but de décrire un environnement tout en estimant les trajectoires des objets en mouvement. Elles répondent au problème du SLAM avec détection et suivi d'objets mobiles.

Le terme SLAMMOT a été introduit dans le travail fondateur de [Wang *et al.*, 2004]. Il s'agit d'une méthode de SLAM 2D par filtrage bayésien reposant sur des données odométriques et télémétriques laser et prenant en compte la présence d'objets mobiles. Les objets sont détectés en segmentant les observations courantes qui ne s'apparient pas avec la carte de l'environnement. La trajectoire de chaque objet est suivie avec un estimateur spécifique.

Les auteurs de [Wangsiripitak et Murray, 2009] proposent une méthode de suivi d'un objet mobile 3D connu, fonctionnant en parallèle d'une méthode de SLAM monoculaire [Davison, 2003]. L'intérêt est de séparer les points appartenant à l'objet et les points fixes de l'environnement pour améliorer la localisation. L'objet est représenté par un modèle 3D. L'approche ne propose pas de méthode pour la détection d'objets mobiles et le modèle de l'objet est prédéfini. L'approche est illustrée figure 1.1.

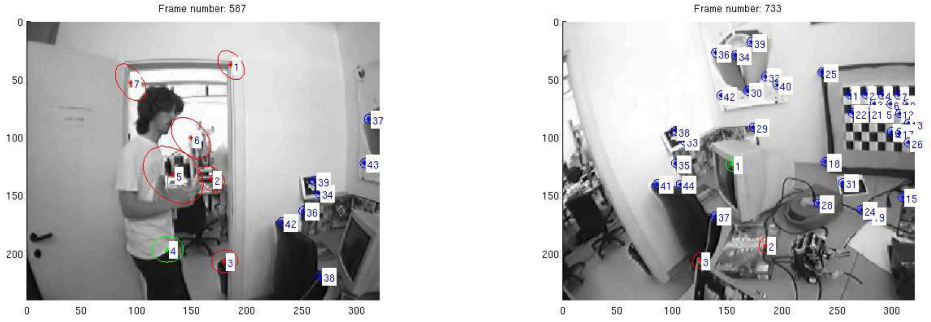


(a) (b)

FIGURE 1.1 – Illustration de l'approche proposée dans [Wangsiripitak et Murray, 2009]. (a) L'objet statique est détecté. (b) L'objet mis en mouvement est détecté et suivi.

Dans l'approche décrite dans [Migliore *et al.*, 2009], les auteurs proposent de classer les points observés comme des points statiques ou dynamiques en utilisant un outil statistique. Ils utilisent plusieurs filtres de Kalman étendu pour modéliser la partie statique de la scène d'une part et suivre les éléments dynamiques d'autre part. Les auteurs admettent néanmoins qu'il est difficile de suivre les objets de manière robuste et montrent des résultats dans un environnement de taille réduite et faiblement dynamique. La figure 1.2 montre les résultats du classifieur de points statiques (en bleu) et dynamiques (en vert) utilisé dans cette méthode sur deux images d'une séquence réelle. Sur la deuxième image, on observe un point classé par erreur comme un point dynamique. Ceci est dû au test de classification basé sur un seuil prédéfini par l'utilisateur.

Les deux approches [Lin et Wang, 2010] et [Marquez-Gamez et Devy, 2012] proposent de répondre au problème en utilisant une paire de caméras stéréoscopiques. La solution proposée par les auteurs de [Lin et Wang, 2010] est d'augmenter le vecteur d'état, formé de la pose de la caméra et des positions des points de l'environnement, avec les états des objets mobiles détectés. Chaque nouveau point détecté est classé en tant que point statique ou dynamique en comparant les résultats du SLAM sous ces deux hypothèses et en retenant l'hypothèse la plus plausible. Le vecteur d'état est mis à jour avec un filtre de Kalman. Les auteurs de [Marquez-Gamez et Devy, 2012] fusionnent une approche reposant sur une grille de mobilité et une approche de classification pour détecter des objets en mouvement. Un vecteur d'état est associé à chaque objet détecté qui est mis à jour par un filtre de Kalman. L'inconvénient de cette méthode est qu'elle ne détecte qu'un certain type d'objet.



(a) (b)

FIGURE 1.2 – Résultats du classifieur de points statiques (en bleu) et dynamiques (en vert) présenté dans [Migliore *et al.*, 2009]. Certains points sont en attente de classification (en rouge).

L'ensemble des méthodes citées précédemment traitent explicitement la présence d'objets mobiles et s'appliquent à estimer les trajectoires des objets en mouvement. L'apparence et la structure des objets n'interviennent cependant pas dans la modélisation de la scène.

Avec l'amélioration des systèmes localisation par vision, de nouvelles approches utilisent l'objet comme une primitive visuelle pour améliorer la précision de la localisation. Elles combinent le SLAM à de la reconnaissance d'objet. Il s'agit en général d'objets appris dans une étape préalable. Les auteurs de [Castle *et al.*, 2010] construisent une base de données d'objets planaires pour les reconnaître en ligne et les utiliser comme points de repère statiques robustes lors de la localisation. Les auteurs de [Botterill *et al.*, 2009] améliorent un algorithme de SLAM métrique avec la reconnaissance d'objets de taille connue. Pour cette méthode, il n'est pas nécessaire que les objets aient une place fixe. Les auteurs de [Kim *et al.*, 2010] présentent une méthode de suivi de multiples objets en temps réel dans un environnement inconnu. Dans toutes ces méthodes, les objets sont définis comme un ensemble de primitives visuelles rigidement liées dont l'apparence est encodée dans un modèle appris hors ligne.

Certaines méthodes proposent aussi à l'utilisateur d'apprendre de nouveaux objets en ligne, *i.e.* pendant la reconstruction de l'environnement. Dans l'approche de [Reitmayr *et al.*, 2007], l'utilisateur doit segmenter manuellement les objets lors d'un premier passage et le système reconnait ces objets lors d'une seconde exploration. De la même façon, les auteurs de [Kim *et al.*, 2010] proposent à l'utilisateur d'enrichir la base données existante en sélectionnant une région d'intérêt dans l'image. Ces méthodes d'apprentissage sont manuelles ou semi-automatiques.

1.4 Conclusion

La modélisation des environnements pour la localisation est un problème difficile, comme peut le montrer la diversité des méthodes présentées dans cet état de l'art. L'ensemble des méthodes présentées dans la section 1.2 enregistrent la structure statique de la scène. Les incohérences dues aux mouvements des objets sont ignorées et filtrées au même titre que les données aberrantes. Les approches de type SLAM-MOT décrites dans la section 1.3 prennent en compte la possibilité d'observer les objets dynamiques et s'attachent à estimer la trajectoire de ces objets. Cependant, les expérimentations mettent en œuvre des objets de petite taille et les objets suivis n'interviennent pas dans la modélisation de l'environnement.

Pourtant, dans les environnements intérieurs, les objets temporaires (comme les livres, les affiches...) sont susceptibles d'être observés plusieurs fois et font partie intégrante de l'environnement. Il nous semble donc important de les modéliser dans la description de la scène.

D'autre part, les mouvements induisent un changement de la structure et de l'apparence de la scène et viennent perturber les méthodes de SLAM métrique et de SLAM topologique. Lorsqu'une grande partie de la scène a bougé, il devient difficile pour une méthode de SLAM métrique de distinguer la structure statique et les structures dynamiques et la reconstruction est altérée. De même, la présence d'objets dynamiques perturbe les algorithmes reposant sur la reconnaissance d'image. Lorsqu'un objet saillant est déplacé dans un lieu différent, ce type d'algorithme va reconnaître l'objet présent dans ce lieu et la reconnaissance est alors faussée.

En définitive, il semble qu'aucune des méthodes citées dans cet état de l'art ne fasse intervenir les objets en mouvement dans la modélisation de la scène. Nous proposons de modéliser la scène explicitement comme une structure statique d'une part et un ensemble d'objets dynamiques d'autre part. Cette approche est présentée en introduction et détaillée dans le chapitre 3. Elle permet de tirer un maximum d'informations à partir de différentes explorations d'une caméra dans une scène faites à des instants différents. D'autre part, elle offre une description cohérente et exploitable dans des applications de localisation. Nous présentons dans le chapitre suivant les deux méthodes de localisation par vision qui fournissent les données d'entrée de notre modélisation. L'approche que nous proposons fait l'objet des chapitres 3 et 4. Le chapitre 5 décrit son application à une méthode de localisation.

Chapitre 2

Algorithmes de vision pour la localisation

Contenu du chapitre

Nous présentons dans ce chapitre deux méthodes de localisation dans un environnement a priori inconnu : un algorithme de SLAM métrique et un algorithme de SLAM topologique. Ces méthodes fournissent les données d'entrée de notre étude. Enfin, nous présentons une méthode d'optimisation robuste et les améliorations que nous lui avons apportées qui seront amplement utilisées dans ces travaux.

2.1 Algorithme de localisation métrique

Nous présentons dans cette section et la section suivante les deux approches de localisation utilisées. La localisation *métrique* fournit une carte des points 3D de l'environnement et la position et l'orientation précises de la caméra au cours du temps. La localisation topologique positionne la caméra dans un graphe dont les nœuds représentent des lieux dans l'environnement et les arêtes une relation entre les lieux (adjacence spatiale ou temporelle, similarité). Ce sont les entrées principales de notre méthode. Dans ces travaux, l'algorithme de localisation métrique utilisé est la méthode de localisation et cartographie simultanées proposée par Mouragnon et al. [Mouragnon *et al.*, 2006]. La méthode permet, en temps réel, de déterminer la trajectoire d'une unique caméra dans un environnement inconnu. Elle est résumée figure 2.1. A chaque instant, la caméra observe des points déjà présents dans la carte 3D. Ces observations permettent de calculer la pose de la caméra. La carte est ensuite mise à jour avec l'ajout de nouveau points 3D dans la carte. Une étape d'initialisation fournit les premiers points 3D.

Nous détaillons ci-dessous les différentes étapes de l'algorithme de Mouragnon et al. [Mouragnon *et al.*, 2006].

2.1.1 Détection de points d'intérêt

Nous nous intéressons ici à la détection automatique de points d'intérêt 2D dans l'image. Cette approche permet de représenter l'image par un ensemble d'informations locales. Les points d'intérêt correspondent à des zones saillantes ou coins de l'image. Les détecteurs de points d'intérêt doivent être suffisamment robustes aux changements de perspective pour pouvoir détecter les mêmes points dans les images présentant des points de vue différents. On parle d'invariance de transformation 2D, par exemple l'invariance à la transformation affine ([Mikolajczyk, 2004]). Nous utilisons le détecteur de Harris-Stephens ([Harris et Stephens, 1988]). Sa répétabilité maximise les chances de détecter les mêmes points dans des images successives [Schmid *et al.*, 2000].

Pour respecter les contraintes temps réel, nous sélectionnons un nombre fixe de points d'intérêt, les points qui sont considérés les plus saillants grâce à l'indice de discrimination délivré par le détecteur utilisé.

Enfin, la détection des points d'intérêt est faite par baquets : l'image est découpée en sous-zones et les points d'intérêt sont recherchés dans chacune de ces zones. Les

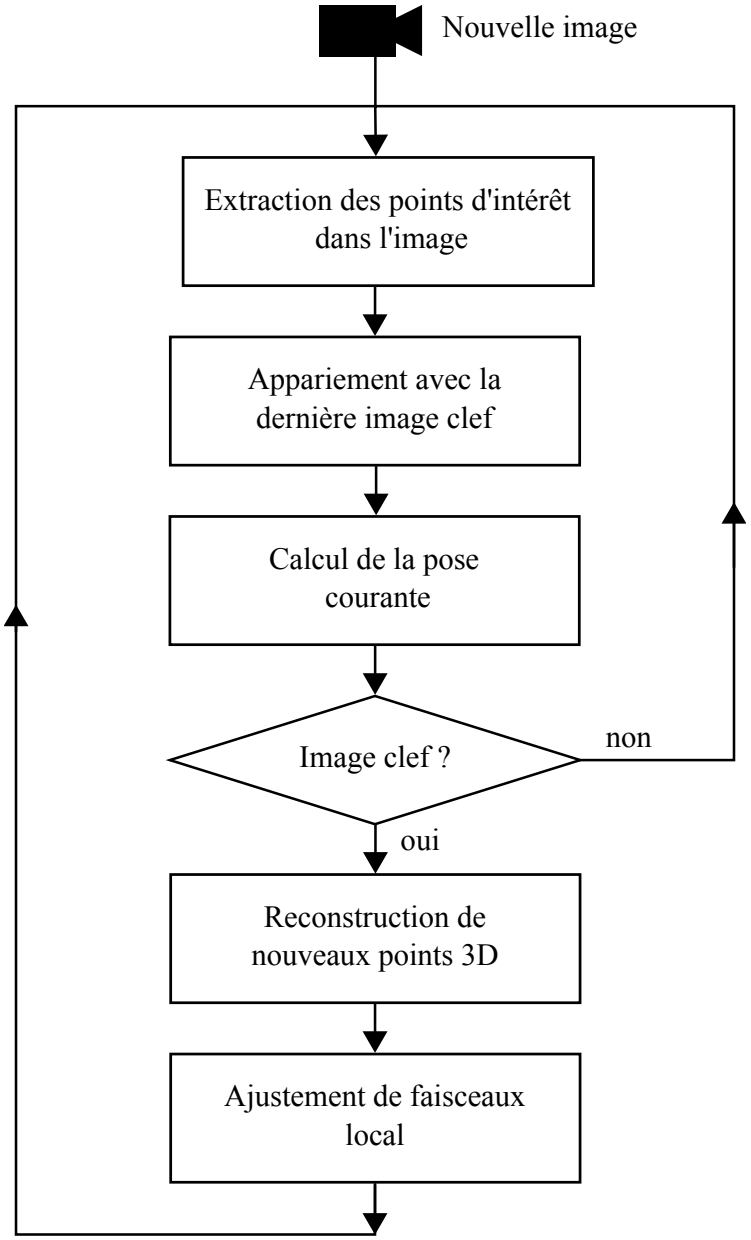


FIGURE 2.1 – Schéma résumant le fonctionnement du SLAM de Mouragnon et al. [Mouragnon *et al.*, 2006] après l'initialisation

points sont bien répartis dans l'image, ce qui permet d'améliorer la précision des résultats des processus de reconstruction 3D.

2.1.2 Description de points d'intérêt

Afin de retrouver les observations d'un même point 3D dans une séquence d'images, nous employons un système de signature sur chaque point d'intérêt. Cette signature est plus communément appelée descripteur. Elle permet de mesurer la similarité entre deux points. Dans les applications de reconstruction 3D et de reconnaissance

de lieu, les descripteurs utilisés doivent être suffisamment invariants aux variations de l'environnement (variations d'illumination) et aux variations dues au mouvement de la caméra (changements de point de vue). Dans notre étude, nous utilisons le descripteur SURF (Speeded Up Robust Features, [Bay *et al.*, 2006]). Dans l'algorithme de SLAM métrique, la variante U-SURF du descripteur est utilisée. Le descripteur U-SURF (U pour *upright*) est rapide à calculer mais il n'est pas invariant à la rotation. Cependant, ses performances sont suffisantes pour apparier les points d'intérêt de deux images consécutives dans un flux vidéo. Pour la reconnaissance de lieu nous utilisons de descripteur SURF invariant à la rotation et au changement d'échelle. Il existe des descripteurs affine-invariants, par exemple ceux proposés par les auteurs de [Lowe, 2004] et de [Guoshen Yu, Jean-Michel Morel, 2011]. Néanmoins, ils ne permettent pas un traitement en temps réel. On peut noter que récemment, deux nouveaux descripteurs, BRIEF [Calonder *et al.*, 2010] et FREAK [Alahi *et al.*, 2012], ont été proposés dans la littérature. Ces descripteurs sont affine-invariants et autorisent des traitements en temps réel.

2.1.3 Appariement des points d'intérêt

Etant donné deux images dont on a extrait les points d'intérêt, il s'agit de faire correspondre les points de la première image avec ceux de la deuxième. L'objectif de cette étape est de retrouver les observations d'un même point 3D parmi l'ensemble des points d'intérêt détectés. A chaque point est associé un descripteur représentant le voisinage du point. Ce descripteur permet de mesurer la similarité du point avec les autres. Pour mettre en correspondance des points, on mesure la distance entre les descripteurs respectifs. Chaque point est alors apparié avec son plus proche voisin dans l'espace du descripteur. Un point ne peut avoir qu'un seul point homologue. Dans la méthode d'origine, un point est apparié à son plus proche voisin selon la distance de corrélation ZNCC (Zero Normalized Cross Correlation). Cette méthode est peu robuste aux changements d'apparence importants, et donc aux larges déplacements de caméra. Nous utilisons descripteur SURF qui repose sur la distribution des ondelettes de Haar 2D (gradient en X et Y) sur le voisinage des points d'intérêt. La distance utilisée est la distance L_1. De plus, pour éliminer une partie des erreurs d'appariements, nous utilisons le critère proposé par Lowe dans [Lowe, 2004]. Un point est apparié à son plus proche voisin si le ratio des distances au plus proche voisin et au deuxième plus proche voisin est inférieur à un certain seuil. Cette stratégie permet de rejeter une grande partie des correspondances ambigües tout en gardant les appariements corrects.

Par ailleurs, nous exploitons aussi le fait que le mouvement relatif des points est assez faible entre deux images successives. On utilise des zones de recherche réduites (40 pixels) autour des points détectés dans l'image précédente. Cette méthode « de proche en proche » limites les erreurs d'appariement et accélère les calculs.

2.1.4 Traitements 3D

Aucune information 3D n'est connue au préalable. Une première étape d'initialisation permet, à partir d'un triplet d'images, de déterminer les poses des premières caméras et de reconstruire les points 3D observés en utilisant l'*algorithme des* 5 *points* [Nistér, 2004]. Une fois cette étape réalisée, les appariements 2D/2D avec les points de la dernière image clef permettent d'associer les observations de la caméra courante avec les points 3D déjà reconstruits. La pose de la caméra est estimée à partir de ces associations 2D/3D. De plus, si l'image est définie comme étant une image clef, elle est utilisée pour mettre à jour la carte de l'environnement. Ces deux étapes sont illustrées sur la figure 2.2.

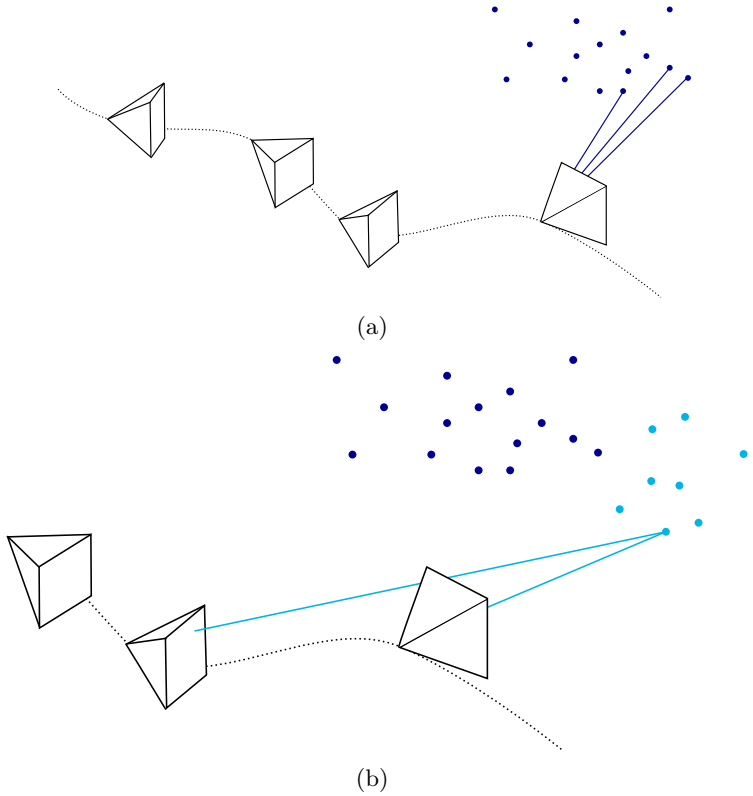


FIGURE 2.2 – **(a) Localisation** : à chaque nouvelle image fournie par le flux vidéo, la caméra est localisée grâce aux associations 3D/2D. **(b) Reconstruction** : si la nouvelle image est une image clef, les points 3D sont reconstruits par triangulation et les paramètres des 3 dernières caméras et des points 3D associés sont optimisés par ajustement de faisceaux

La pose et les observations sont utilisées pour trianguler de nouveaux points et la reconstruction est optimisée par ajustement de faisceaux local. L'ajustement de faisceaux local proposé par [Mouragnon *et al.*, 2006] n'optimise pas l'ensemble des paramètres de la reconstruction. Pour respecter les contraintes de temps-réel, l'algorithme utilise les M dernières caméras et les points 3D associés pour optimiser les paramètres des N dernières caméras. Les $M - N$ dernières caméras sont alors fixées pour garantir la cohérence de la reconstruction. En pratique, $M = 20$ et $N = 3$.

Cette méthode permet d'assurer un traitement en temps réel tout en fournissant une estimation précise des paramètres de la reconstruction.

2.1.5 Notion d'image clef

La méthode de SLAM ne traite pas toutes les images du flux vidéo de la même manière. Chaque image est localisée dans l'environnement reconstruit. Certaines images, appellées images clefs sont utilisées pour mettre à jour la carte des points 3D et lancer une étape d'optimisation par ajustement de faisceaux local. La sélection d'images clefs permet de respecter les contraintes de temps réel. Elle permet également de maximiser les angles de triangulation, et donc d'améliorer la précision des calculs, lors de la reconstruction des points 3D. Le critère de sélection d'une image clef proposé dans [Mouragnon *et al.*, 2006] est le suivant : une image est choisie comme une image clef lorsque le nombre de points en correspondance de l'image courante avec la dernière image clef est inférieur à un certain seuil M. Le nombre de correspondances doit aussi être assez élevé pour garantir un nombre suffisant de points 3D pour le calcul de pose. M est fixé à 400 en pratique par les auteurs de [Mouragnon *et al.*, 2006], ce qui représente un bon compromis.

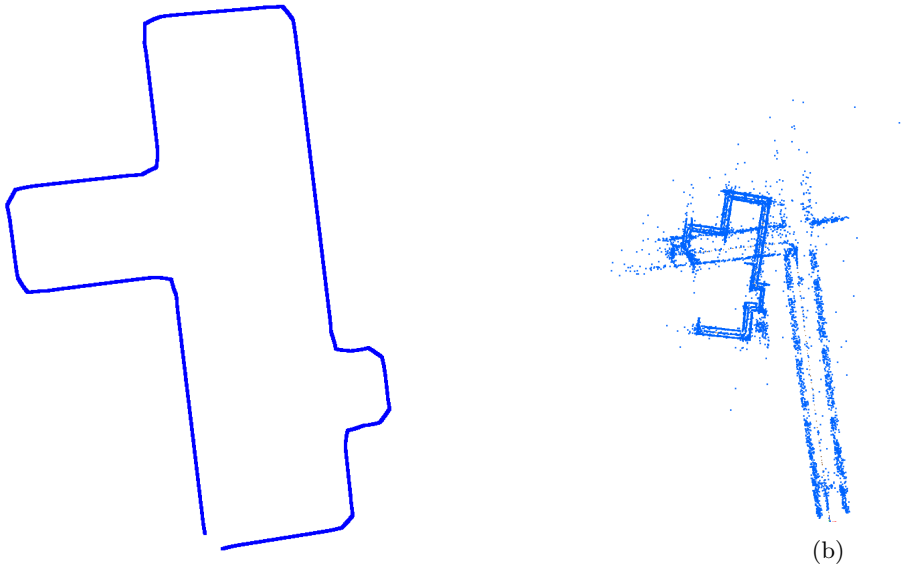
2.1.6 Limites de l'approche

La méthode de SLAM monoculaire présentée dans cette section permet de reconstruire l'environnement et la trajectoire de la caméra de manière précise. Elle présente encore certaines limites que nous listons dans cette section.

2.1.6.1 Dérives sur de longues trajectoires

La performance de l'algorithme présenté dépend aussi de la qualité des données d'entrée. En pratique, les mesures faites sur les images du flux vidéo sont bruitées. Les erreurs et les imprécisions des mesures 2D se répercutent dans l'estimation des points 3D et de la trajectoire. De plus, le processus du SLAM est un processus incrémental : l'estimation des paramètres du modèle à l'instant t s'appuie sur des mesures de l'étape courante et l'historique des estimations jusqu'à $t-1$. Au fur et à mesure, les erreurs s'ajoutent et induisent tôt ou tard une reconstruction incohérente. On parle d'*accumulation d'erreurs* ou de *dérive en position*. Ce problème limite la taille des environnements reconstruits. D'autre part, la reconstruction est réalisée à un facteur d'échelle près, théoriquement constant sur l'ensemble la séquence. On observe en pratique des erreurs sur la norme du déplacement entre la caméra courante et la caméra précédente. Les points sont reconstruits à une mauvaise échelle et cette erreur est transmise de proche en proche aux estimations suivantes. On appelle ce phénomène, la *dérive en facteur d'échelle*.

2.1.6.2 Comportement dans un milieu dynamique



(a) (b)

FIGURE 2.3 – Evaluation de l'algorithme de SLAM métrique dans un environnement dynamique

L'approche de SLAM métrique présenté enregistre uniquement la partie statique de l'environnement. En pratique, les points qui ont un mouvement distinct de celui de la scène sont traités comme des données erronées et filtrés par la procédure RANSAC utilisée lors du calcul de pose. Cependant, lorsqu'une grande partie de la scène bouge, il est difficile de déterminer la structure statique et la reconstruction devient fausse. Nous observons les performances de l'algorithme de SLAM métrique dans un environnement dynamique : l'algorithme est éxécuté sur une séquence de synthèse dans laquelle plusieurs objets en mouvement sont observés. La séquence de 4000 images reconstitue l'exploration d'une caméra mobile dans une ville. Quatre objets planaires sont ajoutés artificiellement sur la trajectoire de la caméra. La trajectoire complète de la caméra est représentée sur la figure 2.3 (a). On compare visuellement la reconstruction 3D de l'environnement obtenue avec l'algorithme de SLAM métrique (figure 2.3 (b)) avec la vérité terrain. On peut observer la déformation de la trajectoire reconstruite, notamment les erreurs du facteur d'échelle. La méthode est fortement perturbée par la présence d'objets dynamiques.

2.2 Reconnaissance de lieu

Nous détaillons dans les parties suivantes une méthode simple et efficace de reconnaissance de lieu. Chaque lieu enregistré est associé à une image de l'environnement. L'image est représentée de façon compacte par un « sac de mots visuels » (section 2.2.1) permettant le calcul rapide d'un score de similarité (section 2.2.3) avec l'ensemble des images d'une base de données. Nous présentons dans la section 2.2.4 des méthodes de validation géométrique permettant de rejeter les fausses détections.

2.2.1 Représentation de l'image

La représentation de l'image utilisée repose sur l'approche proposée par les auteurs de [Nister et Stewenius, 2006]. Comme pour la localisation métrique, les points d'intérêt de l'image sont détectés de manière automatique et à chaque point est associé un descripteur SURF. Si 500 points sont détectés dans l'image, la représentation consiste en un ensemble de 500 vecteurs de 64 composantes. On compresse cette représentation en remplaçant chaque descripteur de l'image par l'index d'un descripteur pris dans un dictionnaire de mots visuels. Le dictionnaire de mots visuels est un ensemble de k descripteurs SURF qui doit représenter au mieux l'espace des descripteurs. L'apprentissage du dictionnaire est expliqué dans la section suivante 2.2.2. On recherche pour chaque descripteur extrait dans l'image son plus proche voisin dans le dictionnaire. La distance utilisée est le produit scalaire entre les vecteurs SURF. On compte les mots du dictionnaire qui sont présents dans l'image. L'image est finalement représentée par un vecteur de k composantes (figure 2.4).

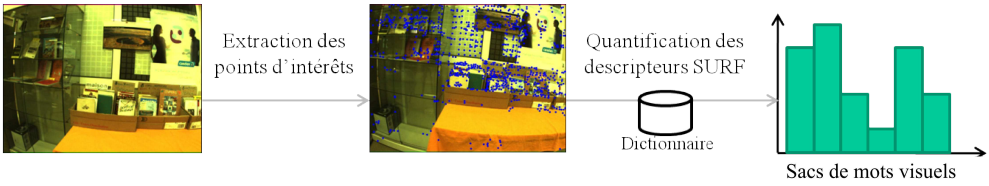


FIGURE 2.4 – Représentation de l'image avec un sac de mots visuels.

2.2.2 Apprentissage du dictionnaire

Le dictionnaire de mots visuels doit décrire au mieux l'espace des descripteurs qui est un espace à 64 dimensions. On génère une famille de k vecteurs SURF en appliquant un algorithme de partionnement de données (*clustering*) sur l'ensemble des descripteurs extraits d'un grand nombre d'images d'apprentissage. L'algorithme utilisé est l'algorithme des k-moyennes (*k-means*). Cette méthode d'apprentissage non supervisé permet de regrouper les descripteurs en k partitions ou *clusters*. Chaque descripteur appartient alors à la partition dont la moyenne est la plus proche. Etant donné $(D_1, ..., D_n)$ l'ensemble des descripteurs extraits dans les images d'apprentissage, l'algorithme k-means regroupe les n descripteurs dans $k \leq n$ partitions $W = \{W_1, ..., W_k\}$ afin de minimiser la somme :

$$\sum_{i=1}^{k} \sum_{D_j \in W_i} \|D_j - w_i\| \tag{2.1}$$

L'ensemble des mots visuels est défini par $\{w_1, ..., w_k\}$, w_i étant la moyenne des descripteurs de W_i. L'algorithme standard choisit initialement k mots au hasard. Il repète ensuite les deux étapes suivantes jusqu'à la convergence :

- assigner chaque descripteur au mot visuel le plus proche
- mettre à jour la valeur du mot visuel (moyenne des descripteurs)

La convergence est atteinte lorsque qu'il n'y a plus de changement. Les auteurs de [Nister et Stewenius, 2006] proposent une construction hiérarchique permettant

d'accélérer les calculs. Le processus est illustré figure 2.5. On construit au départ p cellules ou partitions avec l'algorithme des k-moyennes. Puis, la même procédure est appliquée récursivement à chaque cellule un certain nombre de fois pour atteindre le nombre de k partitions.

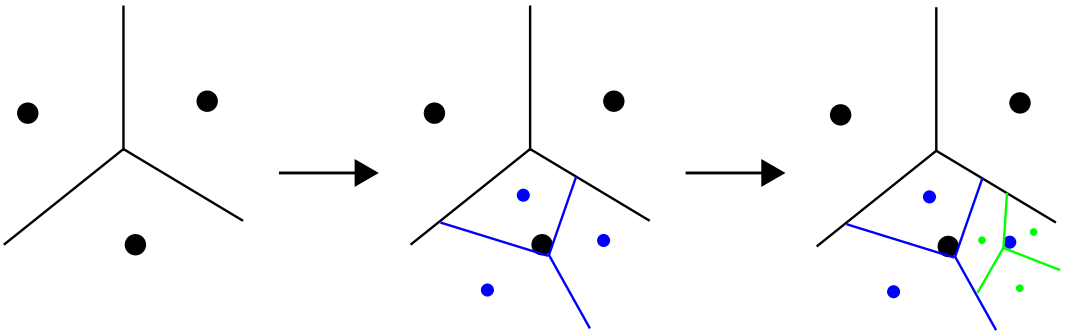


FIGURE 2.5 – Illustration du processus de construction du dictionnaire hiérarchique à 3 branches. A chaque étape, les descripteurs sont regroupés dans 3 cellules, on montre la partition d'une seule cellule pour simplifier le schéma

2.2.3 Définition du score de similarité entre deux images

Une image est associée à un vecteur h de mots visuels, chaque composante h^i étant égale au nombre d'occurrences du mot w_i dans l'image. Etant donné deux images I_1 et I_2 de vecteurs respectifs h_1 et h_2, le score de similarité est donnée par le produit scalaire des deux vecteurs normalisés :

$$s(h_1, h_2) = \frac{h_1^T h_2}{\|h_1\|_2 \|h_2\|_2} \tag{2.2}$$

$\|v\|_2 = \sqrt{v^T v}$ est la norme L_2 de v. les auteurs de [Nister et Stewenius, 2006] montrent qu'on améliore les performances en pondérant les composantes du vecteur par le facteur d'entropie :

$$IDF_i = ln\frac{N}{N_i} \tag{2.3}$$

avec N le nombre d'images dans la base de données et N_i le nombre d'images dans la base contenant le mot w_i. Ce terme est la fréquence de document inverse (*Inverse Document Frequency*) très souvent utilisée en recherche d'information et en fouille de textes. C'est une mesure de l'importance du mot visuel. Les mots apparaissant peu sont plus discriminants et ont un poids plus important. Inversement, les mots fréquents ont un poids très faible. Au final, la composante h_i du sac de mots visuels vaut :

$$TF - IDF_i = \frac{n_i}{n} ln\frac{N}{N_i} \tag{2.4}$$

avec n_i le nombre d'occurrence du mot w_i et n le nombre de mots dans l'image. n_i/n est la fréquence du mot dans l'image (*Term Frequency*).

2.2.4 Vérification de la géométrie

La reconnaissance de lieu nécessite une étape de validation géométrique. En effet, étant donnée une image requête, le score 2.2 défini dans la section précédente permet de trouver parmi les images enregistrées dans la base de données l'image la plus proche qui obtient le plus haut score de similarité. Cependant cette image ne provient pas forcément du même lieu. Deux lieux distincts peuvent avoir la même apparence et on observe souvent des erreurs dues à ce phénomène appelé *aliasing* perceptuel. Pour améliorer la robustesse de la reconnaissance de lieu, on introduit plusieurs méthodes de validations géométriques présentées ci-dessous.

2.2.4.1 Validation géométrique faible

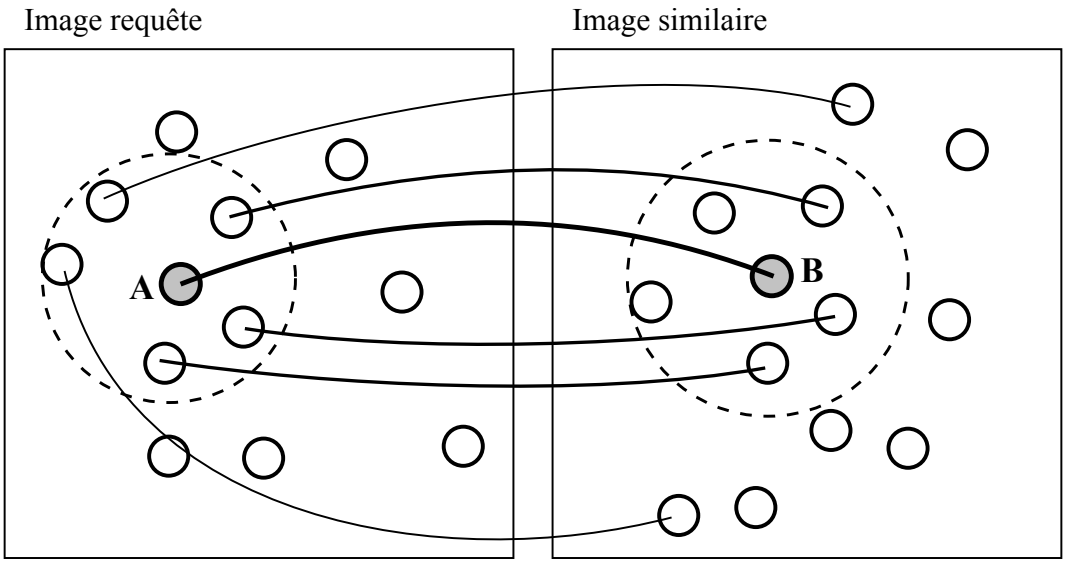


FIGURE 2.6 – **Vérification géométrique faible.** Pour vérifier une correspondance de points (A,B), on compte le nombre de points appariés dans les zones définies par les $k = 5$ points adjacents dans chacune des images. Si ce nombre est nul, la correspondance est rejetée.

La validation géométrique faible consiste en une vérification de la cohérence spatiale des points appariés dans les deux images similaires. La disposition des mots extraits dans l'image de la base de données doit être identique à celle des mots extrait dans l'image requête. L'étape de validation définit pour chaque correspondance de points une zone de recherche dans chacune des images constituée des k points les plus proches spatialement. On compte le nombre de points appariés dans ces deux zones de recherche. Si ce nombre est nul, la correspondance est rejetée. Cette méthode est illustrée figure 2.6. Les scores de similarité donnés par 2.2 sont mis à jour en ajoutant le nombre de vote de cohérence spatiale. Les images de la base de données sont triées à nouveau avec ce score. Les auteurs de [Sivic et Zisserman, 2009] ont démontré les performances de cette méthode dans une application de reconnaissance d'objets. Nous pouvons appliquer cette méthode à la reconnaissance de lieu. Si un nombre minimal de points est validé par cette étape, l'hypothèse est acceptée. Dans le cas

contraire elle est rejetée. La méthode rejète une grande partie des fausses détections mais elle n'est pas aussi précise que la validation géométrique forte présentée dans la section suivante.

2.2.4.2 Validation géométrique forte

Cette méthode calcule précisément la géométrie entre les deux vues. A partir des correspondances de points entre les deux vues, la matrice fondamentale est estimée avec une procédure RANSAC. L'hypothèse de reconnaissance de lieu est validée si le nombre de points vérifiant la transformation dépasse un seuil. Cette méthode est appliquée dans de nombreuses approches de reconnaissance de lieu et de SLAM topologique [Cummins et Newman, 2009], [Angeli *et al.*, 2008]. Elle permet de rejeter une grande partie des fausses détections.

2.2.5 Discussions

La méthode de reconnaissance de lieu présentée dans cette section repose sur le critère de maximum de vraisemblance. On recherche dans une base de données le lieu qui « ressemble » le plus à l'observation courante. Cette méthode a l'avantage d'être simple à mettre en œuvre. Elle repose sur les comparaisons exhaustives de l'observation courante avec l'ensemble des lieux modélisés dans la carte pour en déduire l'hypothèse la plus vraisemblable. L'approche en sac de mots visuels permet de s'adapter aux environnements de grande taille. Le critère de maximum de vraisemblance est néanmoins sensible à l'*aliasing* perceptuel : deux lieux distincts peuvent avoir la même apparence. Une validation géométrique adaptée est indispensable pour rejeter les fausses détections. Cependant, la combinaison du critère de maximum de vraisemblance et de la validation géométrique n'est pas adaptée aux environnements dynamiques. Nous présentons ci-dessous les limites de l'algorithme dans une environnement dynamique.

2.2.5.1 Limites de la méthode dans les milieux dynamiques

Des déplacements dans la scène altèrent l'apparence des lieux. Il arrive alors que des décisions erronées soient prises. Notamment, lorsqu'un lieu n'est pas reconnu à cause du déplacement d'un objet, un nouveau lieu est alors ajouté à tort dans la base de données et le taux de faux positifs augmente. Dans ce cas, deux modèles d'apparence plus ou moins proches sont utilisés pour décrire le même lieu. Avec cette accumulation de modèles, plusieurs d'hypothèses fiables (*i.e.* hypothèses d'un haut score de similarité) coexistent et il est de plus en plus difficile de discerner l'hypothèse la plus vraisemblable. Ce phénomène de surapprentissage perturbe la reconnaissance de lieu. D'autre part, lorsqu'un objet saillant est déplacé dans un lieu différent, la méthode va reconnaître l'objet présent dans ce lieu et la détection est faussée. Nous avons éprouvé cette méthode dans un milieu dynamique. Deux séquences vidéos ont été acquises dans ce lieu. Plusieurs objets ont été déplacés entre

TABLE 2.1 – Cas limites de la méthode de reconnaissance de lieu dans un environnement dynamique

ces deux acquisitions. L'environnement est modélisé avec les images de la première séquence pendant une première phase d'apprentissage. La méthode de reconnaissance de lieu fait correspondre les images de la deuxième séquence avec celle de la première. Le tableau 2.1 présente cinq fausses détections de lieu qui sont toutes dues

à la présence d'objets déplacés dans un lieu différent. On observe à gauche l'image courante et à droite l'image de la base de données. Les points bleus sont les points validés par le calcul de matrice fondamentale entre les deux vues. Les points rouges sont les *outliers*. Nous constatons ici que ces fausses détections ne sont pas rejetées par l'étape de validation géométrique. Le critère de maximum de vraisemblance est donc sujet aux erreurs temporaires de détection : la ressemblance n'est due qu'à la présence de l'objet dans l'image courante. Néanmoins, nous constatons sur ces résultats que cette sensibilité peut être utile pour apprendre de nouveaux objets. Le critère de maximum de vraisemblance est utilisé dans une étape importante de l'algorithme d'extraction automatique d'objet présenté dans le chapitre suivant.

2.3 Optimisation numérique robuste : l'algorithme RANSAC

2.3.1 Présentation de l'algorithme

L'algorithme RANSAC (Random Sample Consensus) est un algorithme robuste permettant d'estimer les paramètres d'un modèle à partir d'un ensemble de données bruitées contenant des valeurs aberrantes. Les données bruitées vérifient le modèle recherché avec une erreur plus ou moins importante. Les données aberrantes (*outliers*) ne correspondent pas au modèle recherché. Elles peuvent venir de mesures erronées ou d'erreurs d'interprétation des données. Le problème est le suivant : il s'agit d'estimer les paramètres $\mathbf{X}$ à partir d'un ensemble de M observations. Il suffit cependant de $m < M$ observations pour déterminer le modèle. Le problème est donc surdéterminé. RANSAC procède en répètant itérativement les trois étapes suivantes.

1. **Génération d'une hypothèse.** Un échantillon de taille minimale est sélectionné en tirant aléatoirement m valeurs parmi les M observations. La taille minimale de l'échantillon permet d'augmenter la probabilité de créer un échantillon dans données aberrantes. Les paramètres du modèle sont calculés à partir de l'échantillon.
2. **Evaluation de l'hypothèse.** On détermine alors l'ensemble des données validant le modèle calculé (*inliers*). Il faut définir pour cela une mesure de distance au modèle et un seuil τ au dessus duquel les données ne sont pas considérées comme acceptables.
3. **Test de la condition d'arrêt** L'algorithme s'arrête si la probabilité d'avoir créé au moins un échantillon non contaminé (i.e. ne contenant aucune donnée aberrante) dépasse un certain seuil. Soient p la probabilité d'avoir sélectionné au moins un échantillon non contaminé, N, m et I respectivement, le nombre total d'observations, le nombre minimal de valeurs pour déterminer le modèle et une estimation du taux d'inliers. Le nombre d'itérations préconisé est donné par :

$$N = \frac{log(1-p)}{log(1-I^m)} \tag{2.5}$$

RANSAC n'est pas une méthode de minimisation comme la méthode des moindres carrés. L'algorithme permet de trouver le modèle permettant d'expliquer le plus

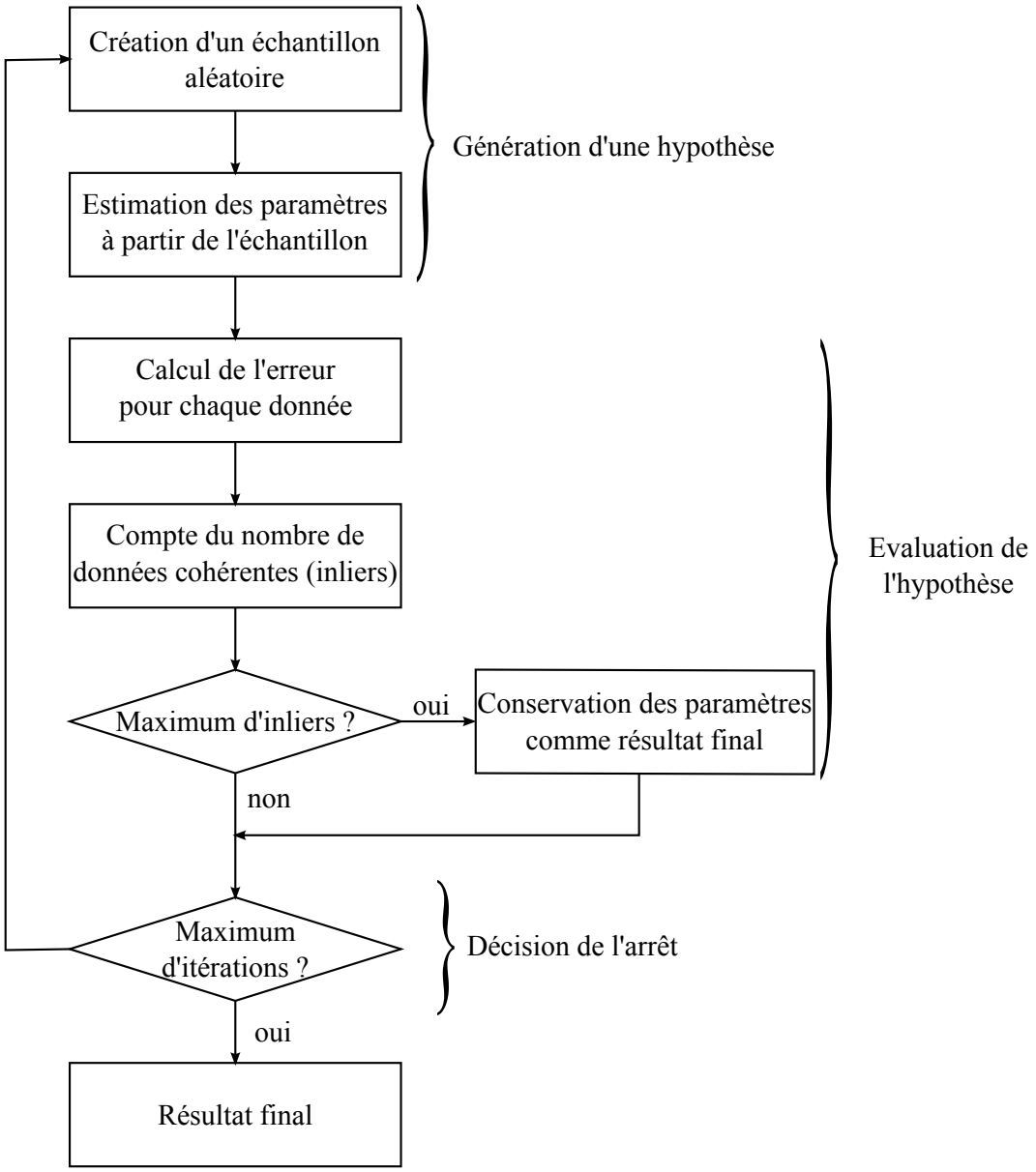


FIGURE 2.7 – **L'algorithme RANSAC.**

grand nombre de données. Le processus est résumé par le diagramme de la figure 2.7.

Prenons l'exemple de l'estimation d'une droite à partir d'un nuage de points en deux dimensions. Chaque point est décrit par ses coordonnées x, y. Le modèle est décrit par l'équation $ax + by + 1 = 0$ et $X = [a, b]$ sont les paramètres à estimer. Seulement $m = 2$ points sont nécessaires pour déterminer a et b. La distance d'un point au modèle est donnée par :

$$distance((x, y), (a, b)) = \frac{|ax + by + 1|}{\sqrt{a^2 + b^2}} \tag{2.6}$$

La figure 2.8 donne trois étapes de la procédure appliquée à l'estimation d'une droite. Les étapes de RANSAC sont les mêmes quelque soit le modèle mathématique

recherché. Seule la taille de l'échantillon formé pour générer les hypothèses change. Dans ces travaux, nous appliquons la procédure à l'estimation d'homographie, de matrice fondamentale et de pose de caméra.

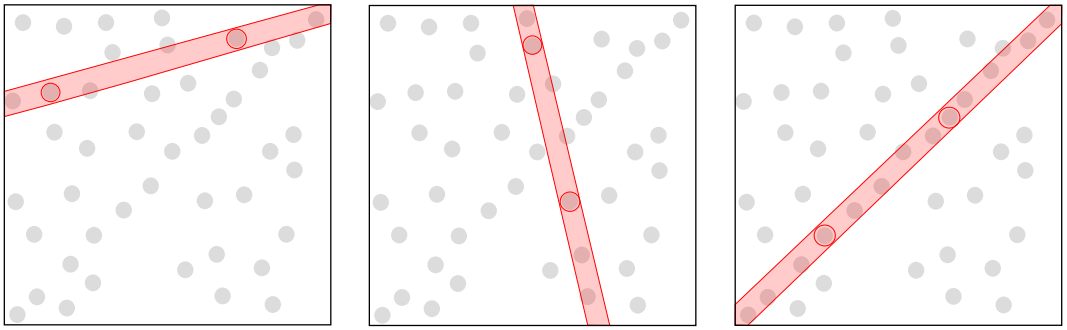


FIGURE 2.8 – L'algorithme RANSAC appliqué à l'estimation d'une droite.

2.3.2 BetaSAC : une nouvelle méthode d'échantillonnage

Chacune des étapes de la procédure RANSAC a fait l'objet de multiples recherches visant à accélérer le processus ou augmenter la précision des résultats. De nombreuses approches proposent une amélioration de l'échantillonnage. Nous détaillons ici les travaux de Meler et al. [Méler *et al.*, 2010].

BetaSAC est une nouvelle méthode d'échantillonnage permettant de créer plus rapidement des échantillons consistants, *i.e.* composés de points (ou de correspondances de points) vérifiant le même modèle. Un échantillon est formé de manière incrémentale et l'ajout d'un point est conditionné par les points déjà présents dans l'échantillon. Nous avons vu (section 2.1.1) que les points d'intérêt sont détectés de manière automatique. Les informations fournies par le détecteur (position du point, orientation, échelle) peuvent être utilisées à bon escient lors de la construction de l'échantillon. Ce propos est illustré sur le problème d'estimation de droite figure 2.9. L'algorithme BetaSAC utilise une information sur l'orientation des points pour former un échantillon. En pratique, l'utilisateur doit définir une distance entre points permettant de trier les points par ordre de préférence en regard de l'échantillon en cours de formation. Dans l'exemple de la figure 2.9 la distance entre deux points est la différence entre leur orientations respectives. Considérant cette distance, plus un point est proche du point de l'échantillon, plus la probabilité de l'ajouter pour compléter l'échantillon est élevée. Nous montrons dans le chapitre 4 comment appliquer cette méthode à l'estimation d'homographie et de matrice fondamentale.

2.3.2.1 Détails de l'algorithme

La génération d'un échantillon s de taille m suivant la stratégie BetaSAC est résumée par l'algorithme 1.

n est une constante. Dans notre étude, n est fixé à 10. On note ici que l'algorithme

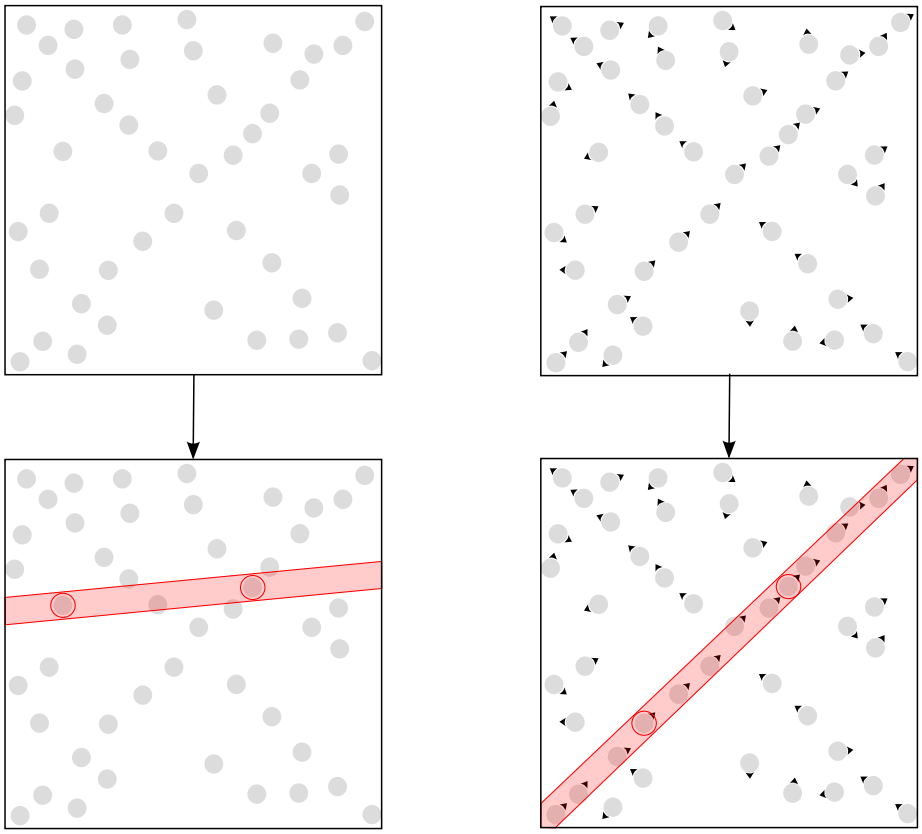


FIGURE 2.9 – Les stratégies RANSAC et BetaSAC appliquées à l'estimation d'une droite à partir d'un ensemble de données bruitées pouvant contenir des erreurs. En gris, les données d'entrée, entourés en rouge les points sélectionnés par la stratégie. Sur la $2^{ème}$ colonne, l'orientation des points est indiquée par une flèche noire. **(a) Stratégie RANSAC :** aucun *a priori* n'est utilisé, l'échantillon contient une donnée erronée. **(b) Stratégie BetaSAC :** Un *a priori* sur l'orientation du point est connu et utilisé, l'échantillon formé est consistant.

ne trie pas l'ensemble des données à chaque fois mais seulement une petite partie. Le temps de calcul additionnel est donc négligeable. Dans l'approche originale, la donnée ajoutée à l'échantillon n'est pas toujours la première du classement. La stratégie de sélection est définie de manière à respecter les propriétés suivantes :

- $\mathcal{P}_1$: chaque échantillon a eu la même chance d'être formé après T itérations,
- $\mathcal{P}_2$: les échantillons les plus prometteurs sont tirés dans les premières itérations.

A l'itération t, l'algorithme calcule le vecteur de sélection $\vec{i}(t) = [i_1(t), ..., i_m(t)]$ de taille m. $i_l(t)$ est le rang dans le classement de la $l^{ième}$ donnée ajoutée à s à l'itération t. La suite de vecteur $\vec{i}(t)$ de déplace dans $\{1, ..., n\}^m$ au cours des itérations. $i_l(t)$ est un nombre généré par la variable aléatoire $B_{i_l(t)/n}$ dont la fonction de densité est définie à partir d'une loi bêta. Il est démontré dans l'article [Méler *et al.*, 2010] que cette variable permet de vérifier les deux propriétés énoncées.

Algorithm 1 Tirage d'un échantillon complet suivant la stratégie BetaSAC

1: s $\leftarrow \emptyset$
2: **for** $l = 1$ to m **do**
3: Tirer aléatoirement n données.
4: Trier les n en fonction de s en utilisant une fonction de tri approprié.
5: d est la première donnée dans le tri.
6: s $\leftarrow$ s $\cup$ d
7: **end for**

2.4 Algorithme de *clustering* : le *mean-shift*

Les méthodes reposant sur le mean-shift sont très populaires en classification de données grâce à leurs performances et leurs faibles coûts de calcul [Comaniciu et Meer, 1998]. Ces méthodes présentent l'avantage de fournir automatiquement le nombre de classes. L'algorithme *mean-shift*, proposé par les auteurs de [Fukunaga et Hostetler, 1975], est une approche non-paramétrique permettant de détecter les modes d'une distribution de probabilité. L'approche considère que les observations dans l'espace des caractéristiques de dimension d sont un échantillonnage d'une fonction de densité de probabilité. Les régions denses dans l'espace des caractéristiques correspondent aux maximums locaux de cette distribution implicite. On procède pour chaque point à une montée de gradient convergeant vers le maximum local. Les maximums obtenus sont les modes de la distribution et les points associés à chaque maximum appartiennent à la même classe (*cluster*).

Le filtrage mean shift est décrit dans [Comaniciu *et al.*, 2002]. Nous rappelons ici les principaux résultats. Etant donné n points x_i, $i = 1, ..., n$ de l'espace des caractéristiques de dimension d $\mathbb{R}^d$, l'estimateur par noyau multivariable utilisant un noyau à symétrie radiale $K(x)$ est donné par :

$$\hat{f_K} = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{2.7}$$

où h est un paramètre définissant la taille du noyau. Le noyau à symétrie radiale est défini de la manière suivante :

$$K(x) = c_{k,d} k(\|x\|^2), \tag{2.8}$$

où c_k est une constance de normalisation et $k(x)$, une fonction définissant le profil du noyau. Le gradient de l'estimateur 2.7 est donné par :

$$\nabla \hat{f}(x) = \underbrace{\frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)\right]}_{\text{terme 1}} \underbrace{\left[\frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]}_{\text{terme 2}}, \tag{2.9}$$

où $g(x) = -k\prime(x)$ est la dérivée du profil du noyau. On définit le noyau $G(x)$ par :

$$G(x) = c_{g,d} g(\|x\|^2), \tag{2.10}$$

avec $c_{g,d}$ la constance de normalisation correspondante. Le premier terme de l'équation 2.9 est proportionnel à la densité estimée au point x calculée avec le noyau G. Le second terme est le vecteur *mean-shift* $\mathbf{m}$, *i.e.* la différence entre la moyenne des points calculée sur une fenêtre de taille h et pondérée par le noyau G et le point x situé au centre de la fenêtre. L'algorithme *mean-shift* pour un point x_k s'exécute de la façon suivante (figure 2.10) :

1. Calcul du vecteur mean-shift $\mathbf{m}(x_k)$.

2. Déplacement du centre de la fenêtre $(x_k) = (x_k) + \mathbf{m}(x_k)$.

3. Itération des étapes 1. et 2. jusqu'à la convergence $(\nabla \hat{f}(x_k) = 0)$.

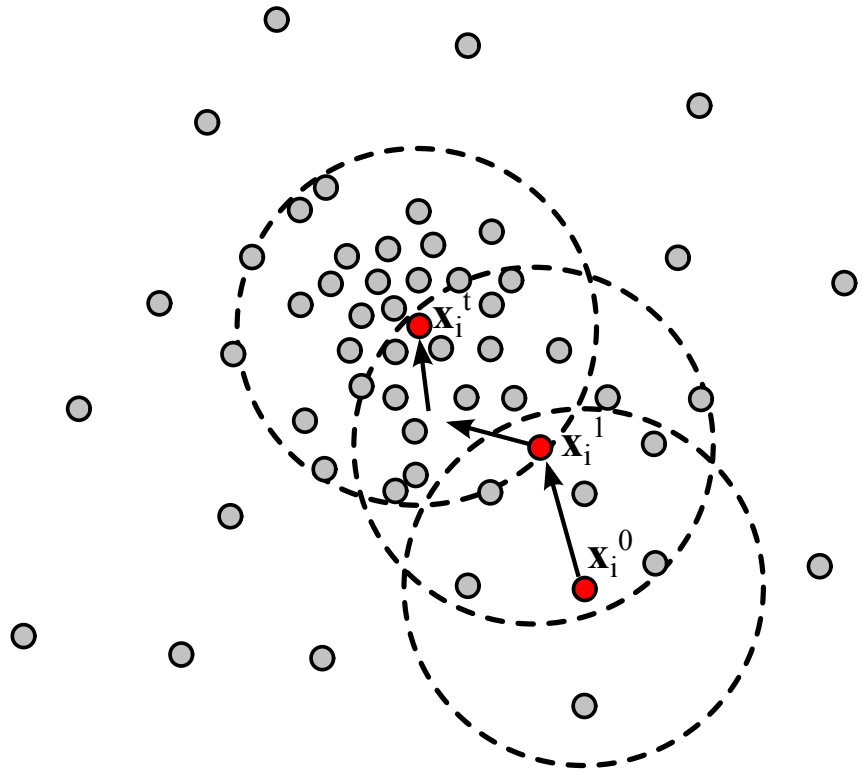


FIGURE 2.10 – **Algorithme *mean-shift*** appliqué au point x_i. L'espace des caractéristiques est un espace de dimension 2. Les données sont représentées sous la forme de points. Les points rouges correspondent aux itérations de l'algorithme $x_i^0, x_i^1, ..., x_i^t$.

Différentes approches ont été proposées pour sélectionner la taille h du noyau automatiquement [Comaniciu *et al.*, 2001], [Singh et Ahuja, 2003].

L'algorithme *mean-shift* est couramment utilisé en vision par ordinateur, notamment pour les problèmes de segmentation d'image [Bradski, 1998], [Comaniciu *et al.*, 2002], de suivi d'objet [Comaniciu *et al.*, 2003] et pour le problème de détection de multiples structures entre deux vues [Wang et Suter, 2004], et [Subbarao et Meer, 2006] que nous étudions dans le chapitre 4.

2.5 Conclusion

Nous avons présenté dans ce chapitre un algorithme métrique de localisation par vision et un algorithme de reconnaissance de lieu reposant sur l'apparence globale de l'image. Ces deux algorithmes fournissent les données d'entrée de la solution proposée pour la découverte automatique d'objets par le mouvement exposée dans le chapitre 3. Nous avons aussi présenté l'algorithme RANSAC, une méthode très répandue en vision par ordinateur pour estimer un modèle à partir d'une ensemble de données bruitées. L'algorithme RANSAC utilisé avec la méthode d'échantillonnage BetaSAC [Méler *et al.*, 2010] est à la base de notre algorithme de détection de multiples modèles présenté dans le chapitre 4.

Chapitre 3

Découverte d'objet par le mouvement

Contenu du chapitre

Nous décrivons dans ce chapitre une nouvelle méthode de découverte automatique d'objets. La méthode proposée compare deux séquences vidéos prises dans une même scène à des instants différents. Elle détecte les changements survenus entre les deux passages et infère la présence d'objet. Enfin elle enregistre une description des objets détectés dans un modèle. Nous définissons en premier lieu les concepts de base utilisés (section 3.1). La section 3.2 donne un état de l'art sur les méthodes d'extraction automatique d'objets existantes. Notre approche est exposée dans la section 3.3 et les expérimentations sont présentées section 3.7.

3.1 Concepts de bases

3.1.1 Découverte d'objet

En robotique, on définit la *découverte* automatique d'objet comme une technique d'apprentissage non supervisée permettant d'apprendre et de modéliser de nouveaux objets à partir d'observations faites dans un environnement *a priori* inconnu ou partiellement connu. Nous étudions dans ce chapitre la capacité de découvrir de nouveaux objets, c'est-à-dire la possibilité de détecter un changement dans l'environnement, d'en déduire la présence d'un objet et de retenir une description exploitable qui permettra de reconnaître cet objet ultérieurement. L'extraction automatique d'objet doit suivre un protocole similaire, en respectant les étapes suivantes :

- **Segmentation** : Le système doit en premier lieu segmenter les objets et les différencier de l'« arrière-plan ».
- **Modélisation** : Le système doit aussi enregistrer les caractéristiques de l'objet dans un modèle.
- **Mise à jour** : Le suivi de l'objet dans l'environnement permet d'améliorer et de mettre à jour le modèle enregistré.

3.1.2 Définitions

Nous proposons une méthode de segmentation par le mouvement visible ou constaté. Les mouvements des objets impliquent des changements dans la scène que l'on peut détecter. Nous définissons, avant toute chose, les trois termes suivants : *environnement*, *structure statique* et *objet*.

L'*environnement* (ou la *scène*) est l'ensemble des lieux explorés par un ou plusieurs utilisateurs. Il est constitué d'une *structure statique* permanente et d'une quantité d'objets rigides pouvant bouger ou être déplacés. La structure statique ne bouge pas par définition. Néanmoins, la modélisation de la structure statique est susceptible d'évoluer au fil du temps grâce aux informations extraites des multiples explorations dans l'environnement. Enfin, dans une définition fonctionnelle, un *objet* est une structure rigide qu'un utilisateur peut prendre et déplacer. L'objet garde une cohérence dans le mouvement. Plus précisément, c'est un ensemble de points 3D ayant un mouvement rigide et cohérent par rapport à la structure statique. L'objet est aussi une structure repérable visuellement. Cette définition implique une notion

de persistance de l'objet. Un objet déplacé peut être observé plusieurs fois dans le même lieu ou dans des lieux différents.

FIGURE 3.1 – Exemple d'environnement semi-dynamique. Certains objets sont susceptibles de bouger (en jaune).

Prenons l'exemple banal d'un bureau, qui est un environnement intérieur exploré quotidiennement. Les murs, le sol et le plafond font et feront toujours partie de la structure statique de l'environnement. Certains meubles fixes (étagères, bureau) sont souvent considérés comme des éléments de la structure statique. Les outils de travail (ordinateur, livres) et les objets personnels (photos, plantes...) sont susceptibles de bouger. Si leur mouvement est détecté, ils sont modélisés en tant qu'objets. Nous utilisons donc une segmentation par le mouvement.

3.1.3 Modélisation des objets

L'objet est représenté et décrit par un modèle. Le modèle enregistre de manière synthétique toutes les informations extraites des observations de l'objet dans l'environnement. Nous utilisons deux types de modèles, largement employés dans le domaine de la vision par ordinateur. Le premier est un modèle d'apparence qui décrit les propriétés visuelles de l'objet. Il s'agit simplement de l'ensemble des primitives visuelles appartenant à l'objet. Ces primitives contiennent des propriétés de couleur, de texture ou de forme et sont détectées dans l'image fournie par le flux vidéo. Le deuxième modèle est un modèle géométrique qui fournit une description de la structure 3D de l'objet. Ce modèle enregistre les positions relatives des points dans l'espace. Il est habituellement détecté en robotique avec un télémètre laser ou une caméra stéréoscopique. Le modèle géométrique que nous enregistrons est éparse. Il est détecté avec un appareil monoculaire en utilisant un algorithme de reconstruction 3D approprié (section 2.1).

Notons ici que nous enregistrons seulement un modèle partiel de l'objet. Nous souhaitons en effet travailler dans de grands environnements et nous utilisons donc des modélisations éparses constituées uniquement de certains points saillants. Ces modèles ne permettent pas de reconstituer l'objet dans son intégralité. Néanmoins,

ils donnent les moyens de détecter sa présence, de le reconnaître et de le suivre dans des images successives de manière précise, voire de le catégoriser et de lui donner une valeur sémantique.

3.2 Méthodes de découverte automatique d'objet

Le fait de pourvoir reconnaître un objet particulier, de suivre son déplacement et de savoir où il se trouve dans la scène offre de nombreuses possibilités d'applications robotiques. Cependant, les systèmes de localisation ne connaissent pas *a priori* tous les objets potentiellement présents dans l'environnement. Ces dernières années, de multiples approches de découverte automatique de nouveaux objets ont été proposées. Nous les classons dans deux catégories : les méthodes de segmentation par l'apparence (section 3.2.1) et les méthodes de segmentation par le mouvement (section 3.2.2).

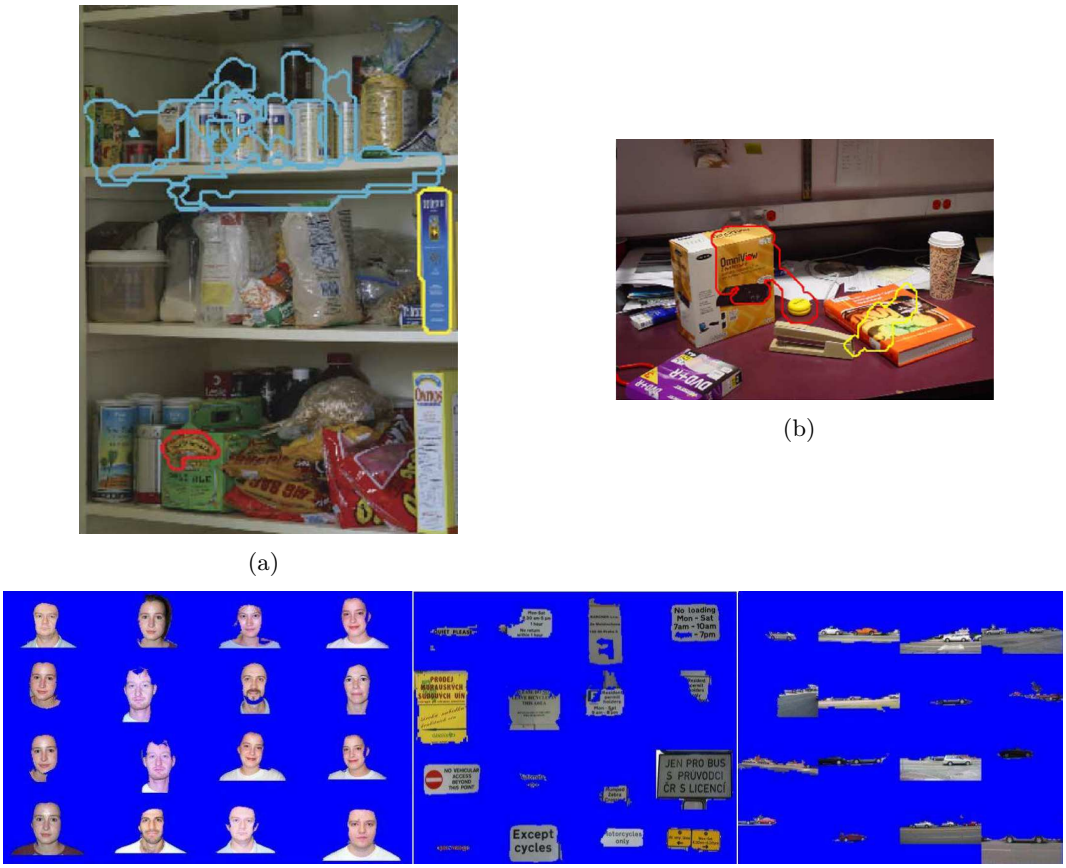
3.2.1 Méthodes de segmentation par l'apparence

Ces méthodes définissent un objet comme un ensemble de primitives cohérentes entre elles (*i.e.* d'apparences similaires) et se distinguant de l'arrière-plan. Les auteurs de [Walther *et al.*, 2004] proposent de segmenter une image en combinant des critères de couleur, d'intensité et d'orientation. Les segments trouvés sont des régions saillantes susceptibles de contenir un objet. Le modèle de l'objet est alors l'ensemble des points SIFT détectés dans un segment.

La segmentation de l'image est aussi utilisée par les auteurs de [Russell *et al.*, 2006] pour découvrir des classes d'objets. Le système segmente une grande collection d'images et classe les segments similaires dans des catégories d'objets. L'objet est défini par l'ensemble des pixels du segment.

Il est évident que ces définitions sont assez floues. Les résultats obtenus dépendent fortement des critères utilisés pour regrouper certaines primitives en objets. Les critères définissent les paramètres de la segmentation et ne sont pas forcément adaptés à tout type d'objet. De plus, ces systèmes ne permettent d'avoir qu'un modèle d'apparence de l'objet. Des exemples d'objets obtenus avec ces deux approches sont présentés figure 3.2 : l'objet n'est pas extrait de manière précise mais le modèle enregistré permet tout de même de le reconnaître dans un lieu différent.

D'autres approches utilisent des algorithmes de reconstruction 3D pour extraire de nouveaux objets. Les auteurs de [Meger *et al.*, 2007] reconstruisent un environnement en 3D en utilisant des caméras stéréoscopiques. Ils combinent l'analyse de la carte de profondeur obtenue avec un détecteur de régions d'intérêt appliqué à l'image (le détecteur MSER de [Matas *et al.*, 2002]) pour extraire des régions saillantes. Un objet est une structure qui se « détache » du fond et du sol. Le modèle enregistre les points SIFT détectés dans la région MSER.



(a) (b) (c)

FIGURE 3.2 – Exemples d'objets obtenus par des méthodes de segmentation par l'apparence. **(a) et (b)** Exemples d'objets trouvés par la méthode [Walther *et al.*, 2004] (délimités par des contours colorés). **(c)** Exemple de classes d'objets trouvés par la méthode [Russell *et al.*, 2006] dans une grande collection d'images.

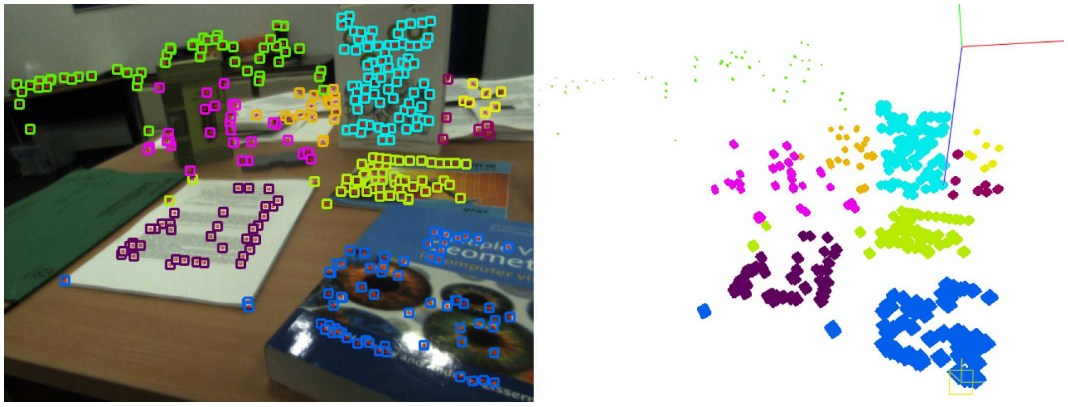


FIGURE 3.3 – Amas de points 3D obtenus avec la méthode de Angeli et al. [Angeli et Davison, 2010]. Chaque objet est un ensemble de points 3D d'une même couleur.

Plus récemment, Angeli et al. [Angeli et Davison, 2010] ont proposé un système permettant de reconstruire l'environnement et de regrouper des points 3D suivant leur similarité et leur proximité spatiale en temps réel. Les amas obtenus sont souvent associés à des objets réels de la scène mais ils ne permettent pas de définir un objet de manière robuste. Néanmoins leur approche permet d'enregistrer un modèle d'apparence et un modèle géométrique 3D de l'objet.

3.2.2 Méthodes de segmentation par le mouvement

Dans la définition donnée en section 3.1.2, l'idée qu'une structure puisse bouger est inhérente à son status d'objet. Deux approches permettent de détecter un tel objet :

- La comparaison de deux observations d'une même scène prises à des instants différents lorsqu'un ou plusieurs objets ont bougé entre les deux instants. Dans ce cas, l'objet n'a pas été vu en déplacement mais ce dernier a été constaté.
- Le suivi d'un objet dans son mouvement.

On trouve dans la littérature plusieurs méthodes proposant ce type d'approche et mettant en œuvre différents capteurs habituellement utilisés en robotique. Ces méthodes sont répertoriées dans le tableau 3.1. Une première idée est d'utiliser une méthode de soustraction de fond entre les images successives d'un flux vidéo. Cette idée est développée dans [Cucchiara *et al.*, 2003], la méthode implique l'utilisation d'une caméra fixe et donne une représentation 2D de l'objet.

Les systèmes proposés dans [Biswas *et al.*, 2002] et [Modayil et Kuipers, 2004] détectent des objets en comparant des grilles d'occupation 2D (*Occupancy grid map*) construites avec l'utilisation d'un sonar ou d'un laser. Les auteurs proposent de classer les observations en observations statiques ou dynamiques. Les observations dynamiques sont ensuite regroupées par des critères de proximité spatiale pour former un objet. Les objets obtenus sont suivis dans les observations suivantes.

Les auteurs de [Southey et Figures, 2006] combinent un algorithme de reconstruction 3D à un algorithme de segmentation pour générer les modèles 3D d'objets

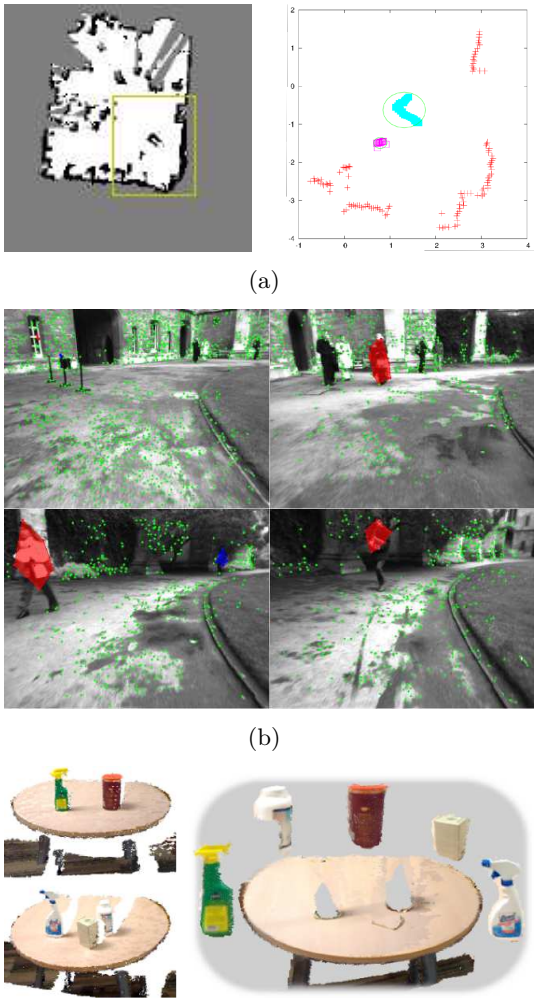


FIGURE 3.4 – Objets obtenus par des méthodes de segmentation par le mouvement. **(a)** Methode de [Modayil et Kuipers, 2004]. A gauche, la carte d'occupation 2D. A droite, les croix rouges indiquent la structure statique et les carrés de couleur sont les objets détectés. **(b)** Méthode de [Kundu *et al.*, 2010], en vert les points statiques, en rouge et bleu des objets détectés et suivis. **(c)** Méthode [Herbst *et al.*, 2011]. A gauche, les deux scènes 3D comparées. A droite, les objets détectés.

en mouvement. L'environnement est reconstruit en 3D à partir du signal d'une caméra stéréoscopique. L'analyse de la carte de profondeur et de la segmentation de l'image permet de générer des hypothèses d'objets en formant des groupes (ou *clusters*) de points 3D. Une hypothèse est retenue si les points 3D du groupe ont un mouvement rigide (*i.e.* la distance entre les positions des points reste constante).

Avec une motivation similaire, Beuter et al. [Beuter *et al.*, 2010] proposent d'analyser des données provenant d'une caméra 3D (caméra munie d'un laser proche de l'infrarouge). Des *clusters* sont formés par des critères de proximité spatiale et de vitesses homogènes et suivis dans les images successives avec un filtre particulaire.

Publications	Méthodes	Capteurs	Modélisation
Cucchiara et al. [Cucchiara *et al.*, 2003]	Soustraction de fond	Caméra monoculaire fixe	Ensemble de pixels
Biswas et al. [Biswas *et al.*, 2002]	Comparaison de scène 2D	Sonar	Grille d'occupation 2D
Modayil et Kuipers [Modayil et Kuipers, 2004]	Comparaison de scène 2D	Laser	Groupe de points 2D sur des observations successives
Southey et al. [Southey et Figures, 2006]	Suivi d'objet 3D en mouvement	Caméras stéréoscopiques	Groupe de points 3D
Beuter et al. [Beuter *et al.*, 2010]	Suivi d'objet 3D en mouvement	Caméra infrarouge	Groupe de points 3D
Kundu et al. [Kundu *et al.*, 2010]	Suivi d'objet 3D en mouvement	Caméra monoculaire	Groupe de points 3D
Herbst et al. [Herbst *et al.*, 2011]	Comparaison de scène 3D dense	RGB-D	Groupe de points 3D

TABLE 3.1 – Etat de l'art des méthodes proposant une segmentation d'objet par le mouvement

Kundu et al. [Kundu *et al.*, 2010] proposent un algorithme de localisation et cartographie 3D par vision monoculaire avec suivi d'objets mobiles. Les objets sont des ensembles de points proches et gardant une cohérence dans le mouvement, néanmoins la structure n'est pas forcément rigide. Le système proposé détecte des objets de petites tailles et utilise le suivi d'objet pour améliorer l'algorithme de reconstruction.

Enfin de récentes études [Herbst *et al.*, 2011, Mason *et al.*, 2012] utilisent des capteurs RGB-D (Kinect) pour détecter des objets qui apparaissent ou disparaissent dans une scène 3D. Ces méthodes nécessitent des reconstructions très précises de l'environnement et ont prouvé leur efficacité dans des environnements de petites tailles.

3.3 Approche proposée

3.3.1 Vue d'ensemble de l'algorithme

Nous proposons un algorithme d'apprentissage automatique d'objets à partir de la comparaison de multiples séquences vidéo prises dans un même environnement. L'algorithme compare les séquences deux à deux.. La méthode suivie est schématisé

figure 3.5. Nous supposons qu'une première exploration a été faite dans l'environne-

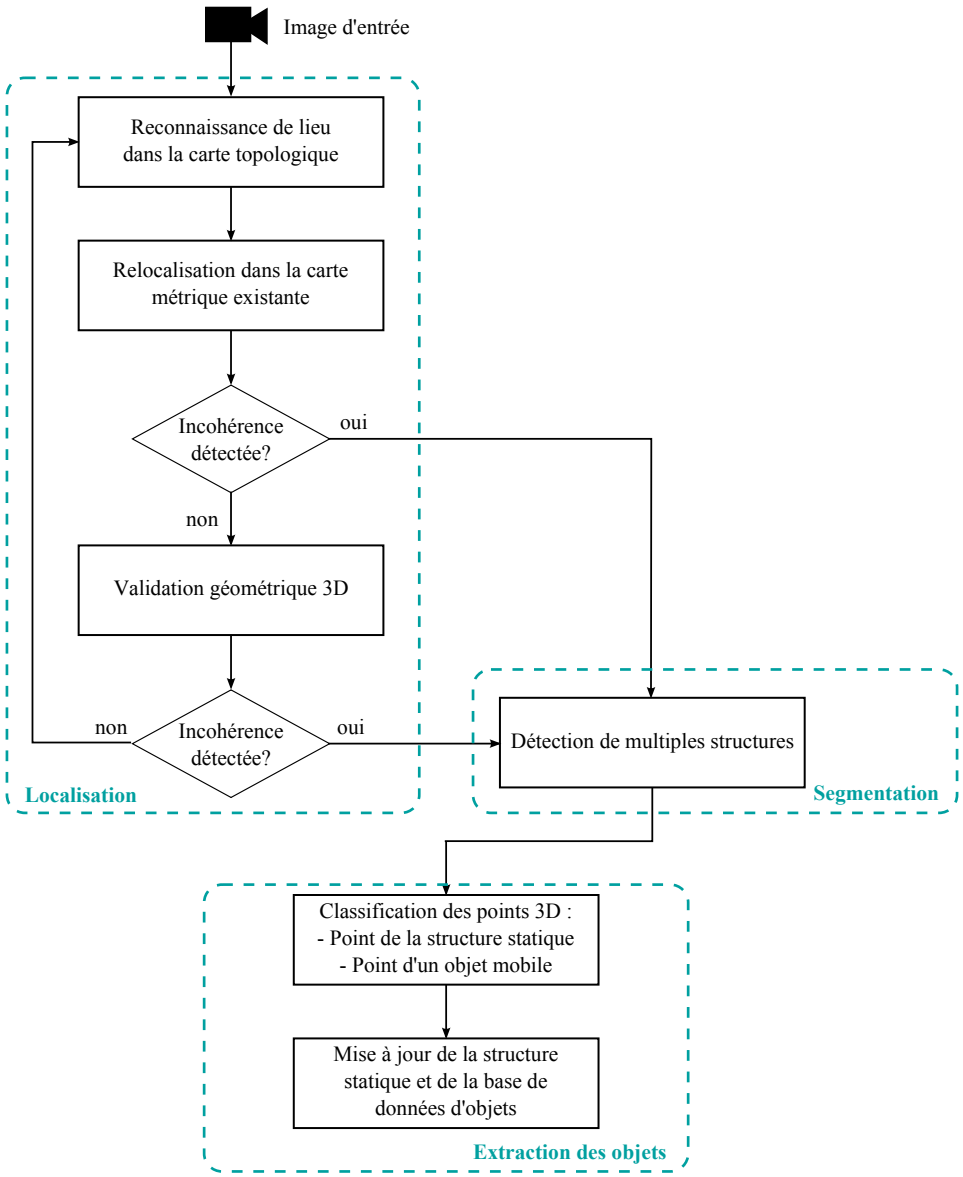


FIGURE 3.5 – **Vue d'ensemble de l'algorithme de découverte automatique d'objet.** La comparaison de deux séquences vidéos permet d'extraire de nouveaux objets.

ment. Nous disposons donc d'une carte topologique construite à partir de la première séquence vidéo. Le traitement de la deuxième vidéo peut être résumé en trois étapes :

1. Reconnaissance d'un lieu présent dans la carte topologique et relocalisation dans la carte métrique existante.

2. Segmentation en différentes structures.

3. Caractérisation de la structure statique et des objets mobiles grâce à l'information 3D.

3.3.2 De la vidéo au modèle

Chaque séquence vidéo permet de construire une carte métrique et une carte topologique de la scène. Nous utilisons l'algorithme de Mouragnon et al. [Mouragnon *et al.*, 2006] pour reconstruire la scène en 3D et déterminer la trajectoire de la caméra. La méthode fournit une carte éparse des points 3D de l'environnement. Comme nous l'avons détaillé dans la section 2.1, elle est sensible à l'accumulation d'erreurs et à la dérive du facteur d'échelle.

Afin de travailler dans de grands environnements, nous enregistrons à la fin de la reconstruction une carte topologique de la scène. L'ensemble des images clefs sélectionnées servent de base d'apprentissage à un dictionnaire de mots visuels. Les descripteurs de chaque image clef sont projetés sur leur plus proche voisin dans le dictionnaire. L'histogramme de mots visuels obtenu est enregistré dans la carte topologique. C'est le modèle d'apparence de l'image.

La scène est modélisée par un graphe. Chaque noeud du graphe enregistre le modèle d'apparence de l'image clef associée. Nous notons dans les arêtes le nombre de points 3D observables dans les deux images clefs. Si les deux images observent deux lieux parfaitement distincts, la valeur de l'arête est nulle. Au contraire, deux images clefs successives ont un grand nombre de points 3D en commun. Le graphe permet ainsi de savoir si deux images sont spatialement adjacentes.

3.4 Relocalisation de l'image courante dans la carte topologique

Nous détaillons dans cette partie l'étape de relocalisation. La méthode met en oeuvre un algorithme classique de reconnaissance d'image et une vérification géométrique utilisant les points 3D reconstruits. Elle permet de comparer la scène 3D reconstruite à partir de la première séquence aux observations provenant de la deuxième séquence et de mettre en lumière des incohérences dans les cartes métrique et topologique.

3.4.1 Reconnaissance de lieu

Nous considérons qu'une première exploration a été effectuée dans l'environnement. Nous avons à disposition une carte topologique de la scène. L'algorithme de localisation et cartographie simultanées est appliqué à une deuxième séquence vidéo prise dans le même environnement. Pour chaque nouvelle image clef, nous recherchons l'image la plus proche dans la carte topologique, en utilisant l'algorithme détaillé en section 2.2. Les descripteurs de la nouvelle image clef sont quantifiés sur le dictionnaire appris sur la première séquence vidéo. On calcule, pour chaque image enregistrée dans la carte topologique, le score de similarité avec l'image courante en utilisant la formule 2.2. L'image la plus proche est l'image qui obtient le meilleur

score, à savoir le score le plus élevé.

3.4.2 Vérification géométrique 2D/3D

L'algorithme de reconnaissance de lieu trouve dans la base de données l'image la plus proche de l'image requête. Cependant, deux images peuvent avoir une apparence similaire sans pour autant provenir du même lieu. C'est le phénomène d'*aliasing* perceptuel. Les hypothèses obtenues avec la reconnaissance de lieu doivent être vérifiées avec une étape supplémentaire faisant intervenir des calculs géométriques. Nous récupérons les points 2D détectés dans l'image similaire et associés à des points 3D de la première reconstruction. Ces points sont mis en correspondance avec les points 2D détectés dans l'image courante en utilisant la méthode détaillée dans 2.1.3. Cependant, la zone de recherche des points homologues est étendue à l'image entière. Les associations 2D/3D formées permettent de calculer la pose relative de l'image courante par rapport à l'image de la base de données. Si la pose peut être calculée de manière précise l'hypothèse de reconnaissance de lieu est validée. En pratique, nous validons une hypothèse si plus de 80% des associations 3D/2D vérifient la pose calculée.

3.5 Détection des incohérences

La relocalisation de l'image courante dans la carte topologique permet de détecter deux types d'incohérences : les incohérences avec la carte topologique et les incohérences avec la carte métrique reconstruite. Nous présentons dans cette section les deux types d'incohérences détectées et l'analyse préalable nécessaire à la caractérisation de la structure statique et des objets dynamiques.

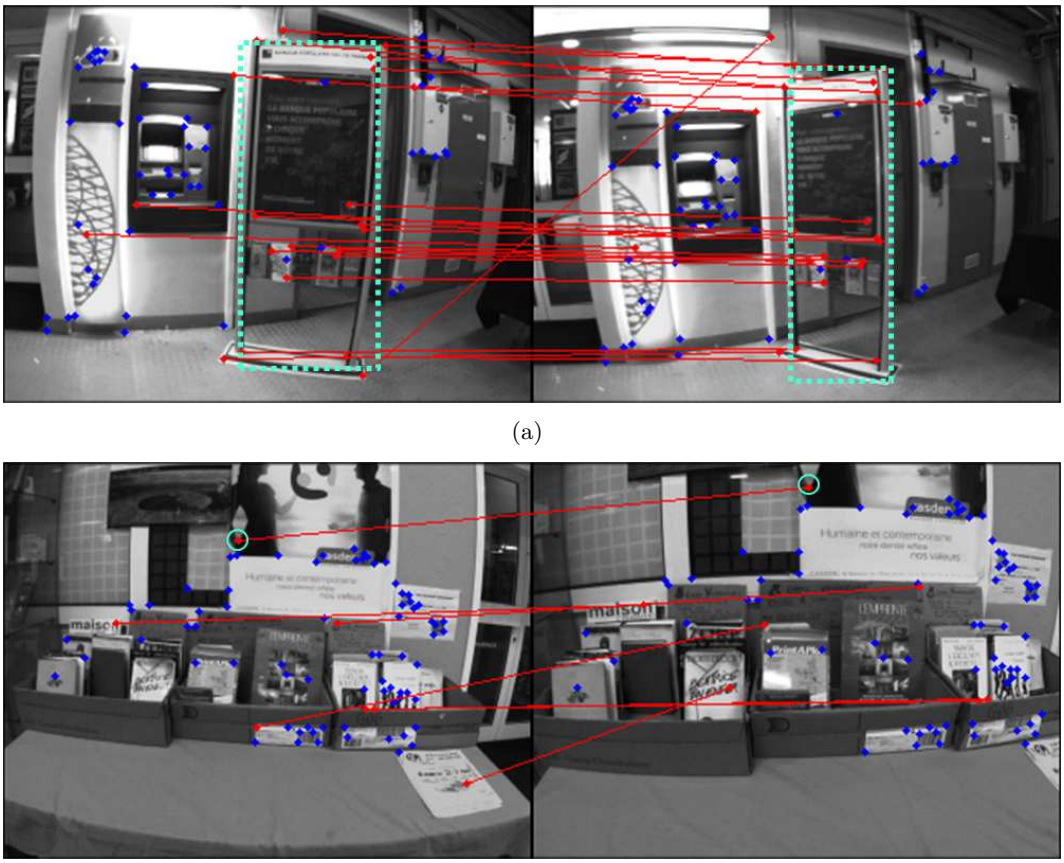
3.5.1 Incohérences avec la carte topologique

Nous supposons que nous avons correctement reconnu un lieu déjà visité, *i.e.* l'hypothèse de reconnaissance de lieu a été validée géométriquement. Une incohérence est détectée si, à l'image suivante, le lieu hypothétiquement reconnu n'est pas relié topologiquement avec le lieu précédemment reconnu. Cette incohérence peut avoir plusieurs causes :

- soit le lieu en question est inconnu et l'hypothèse de reconnaissance est une erreur due à l'aliasing perceptuel,
- soit l'algorithme reconnait un ou plusieurs objets qui étaient initialement dans un lieu différent.

Dans le premier cas, nous ne pouvons pas trouver de modèle géométrique pertinent reliant les deux vues et l'hypothèse est rejetée. Dans le second cas, les objets saillants déplacés dans un lieu différent perturbent la reconnaissance de lieu. Les deux images sont similaires car ce sont les observations d'un ou plusieurs objets dans deux lieux différents. Un algorithme de détection de multiples structures permet d'analyser les correspondances de points entre les deux vues et de segmenter les objets déplacés.

3.5.2 Incohérences dans la carte métrique



(a)

(b)

FIGURE 3.6 – Points 3D incohérents (rouge) mis en évidence par le calcul de pose.

La vérification géométrique 3D/2D permet de confirmer ou d'infirmer une hypothèse de reconnaissance de lieu. Par ailleurs, elle met en évidence la présence de points incohérents avec le calcul de pose. Ces incohérences peuvent être le résultat d'erreurs de mise en correspondance ou provenir d'un manque de précision du modèle calculé. Dans notre contexte, elles peuvent aussi être expliquées par la présence d'un objet ayant bougé entre les deux passages de la caméra. Les points 3D de l'objet déplacé sont incohérents avec la géométrie globale de la scène et viennent perturber le calcul de pose.

La figure 3.6 illustre ces différents cas sur deux paires d'images similaires. L'image de droite est l'image courante relocalisée. Les points affichés sont les points d'intérêt 2D extraits dans l'image. L'image de gauche est l'image similaire de la base de données. Les points affichés sont les projections de points 3D de la première reconstruction observés dans l'image. L'appariement de ces deux ensembles de points permet de calculer la position et l'orientation de l'image courante dans la première reconstruction. Les points bleus sont des points cohérents avec le calcul de pose (*inliers*) et les points rouges sont des points incohérents (*outliers*). Les correspondances de points sont affichées pour les points incohérents. Sur la figure 3.6 (a), les incohé-

rences proviennent de la présence d'un objet qui a bougé dans la scène. La plupart des points 3D incohérents appartiennent à l'objet en question, dont le contour est surligné en pointillés verts.

La figure 3.6 (b) présente un cas limite. Les incohérences proviennent en grande partie des erreurs d'appariement. Néanmoins, une paire de points homologues (points entourés en vert) est jugée incohérente alors que l'appariement semble correct. Ceci peut être expliqué par la difficulté du problème : la différence de point de vue entre les deux images est importante et le modèle calculé n'est pas précis. La moyenne de l'erreur de reprojection est de 1.3 pixels pour les deux images de la figure 3.6 (a). Elle est habituellement inférieure à 0.5 pour un calcul de pose entre deux images clefs successives. Dans ce cas de figure, le seuil délimitant les *inliers* des *outliers* est difficile a définir.

Le calcul de pose permet donc de classer les points 3D en deux catégories : l'ensemble des points cohérents et l'ensemble des points incohérents. Cette étape ne suffit pas à définir un objet. Nous devons vérifier que les points incohérents ne proviennent pas d'erreur d'appariement et que l'ensemble des points suit un mouvement rigide. Dans ce cas, nous pouvons déterminer la présence d'un objet en respectant la définition donnée en début de chapitre, section 3.1.2. Nous résolvons ces problèmes avec un algorithme de détection de multiples structures.

3.5.3 Analyse des incohérences par la détection de multiples structures

L'analyse approfondie d'une incohérence doit répondre aux attentes suivantes :
- vérifier qu'il s'agit soit d'une reconnaissance de lieu dans lequel un ou plusieurs objets ont bougé, soit d'une reconnaissance d'un ou plusieurs objets présents dans un lieu différent,
- segmenter les observations en structures distinctes qui peuvent être classées en structure dynamique ou structure statique.

Nous avons vu dans la section 3.5 que la relocalisation d'une image dans une scène 3D met en évidence un ensemble de points incohérents. Ces points ne proviennent pas nécessairement d'un objet en mouvement. Pour caractériser un objet suivant la définition donnée en section 3.1.2, l'ensemble des points doit avoir un mouvement cohérent. Une première idée est de comparer directement deux scènes en utilisant uniquement les points 3D et non l'ensemble des points 2D détectés dans chaque image. Les objets sont les ensembles de points 3D vérifiant une transformation rigide qui est la composition d'une rotation et d'une translation. La distance entre chaque point de l'objet doit rester fixe. Cependant, les erreurs dans les différentes reconstructions rend le recalage de deux scènes 3D difficile. De plus, les reconstructions sont éparses et peu de points sont reconstruits sur les objets qui peuvent être de petite taille. La figure 3.7 montre la comparaison de deux images d'un lieu dans lequel trois objets ont bougé. Les points incohérents sont déterminés par le calcul de pose (points rouges). Les objets ont respectivement 3, 4 et 8 points reconstruits. Dans ces conditions, il est ardu de caractériser un objet de manière robuste avec cette approche.

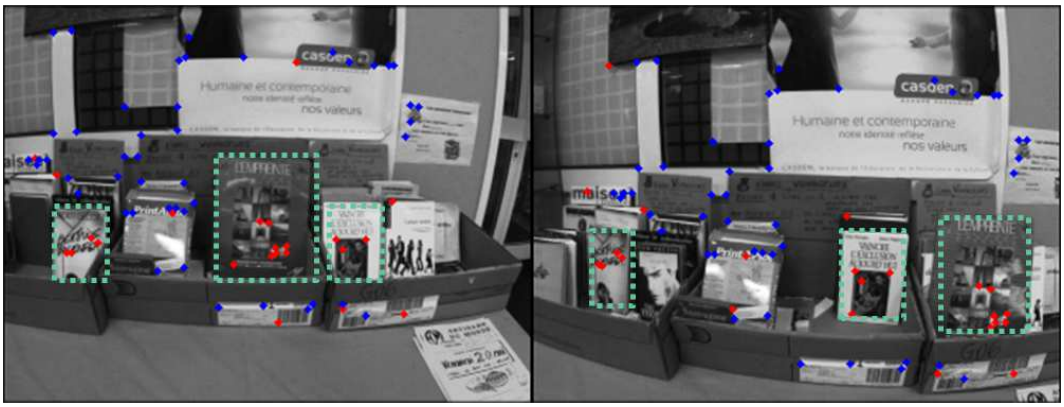


FIGURE 3.7 – Points 3D incohérents appartenant à des objets déplacés, dont le contour est surligné en vert

L'approche retenue, illustrée figure 3.9, est la détection des différentes structures présentes dans la scène à partir des correspondances de points 2D extraits dans les deux images similaires. Il y a en effet beaucoup plus de correspondances de points 2D que de points 3D reconstruits. Ces structures seront ensuite classées en structure dynamique ou structure statique (voir section 3.6).

Nous présenterons dans ce chapitre deux algorithmes permettant de détecter les différents mouvements présents dans la scène. En premier lieu, nous avons limité notre étude à des objets planaires. Il s'agit dans ce cas de trouver, à partir des observations 2D, les différentes homographies entre les deux vues. Nous avons étendu nos travaux à la détection d'objet quelconque. Nous appliquons alors la détection de multiples structure à l'estimation de matrices fondamentales. Dans les deux cas, les ensembles de points trouvés par l'algorithme vérifient une transformation géométrique précise Les points incohérents provenant des erreurs d'appariement sont filtrés.

La détection de multiples structures entre deux vues similaires est étudié dans le chapitre 4. Pour résoudre ce problème, nous devons simultanément regrouper les données selon leur appartenance à un même modèle et estimer les paramètres des modèles ainsi que leur nombre. De plus, le calcul de chaque modèle est perturbé par la présence de bruit, d'erreurs d'appariements et de points vérifiant des modèles différents (les points *pseudo-outliers*). Néanmoins, l'algorithme présenté dans le chapitre 4 permet de résoudre ce problème. Nous verrons que l'estimation de multiples homographies donne de meilleurs résultats que l'estimation de multiples matrices fondamentales. Nous préférons donc utiliser la première approche. Les objets recherchés sont approximés par des plans ou par des ensembles de plans.

3.6 Caractérisation de la structure statique et des objets dynamiques

La validation géométrique, et plus précisément le calcul de pose relative entre deux images similaires, met en évidence la présence de points incohérents. Comme nous l'avons vu dans la partie 3.5, ces points ne respectent pas le mouvement global de la scène. Ces incohérences proviennent soit d'erreurs d'appariement, soit du mouvement d'un objet dans la scène.

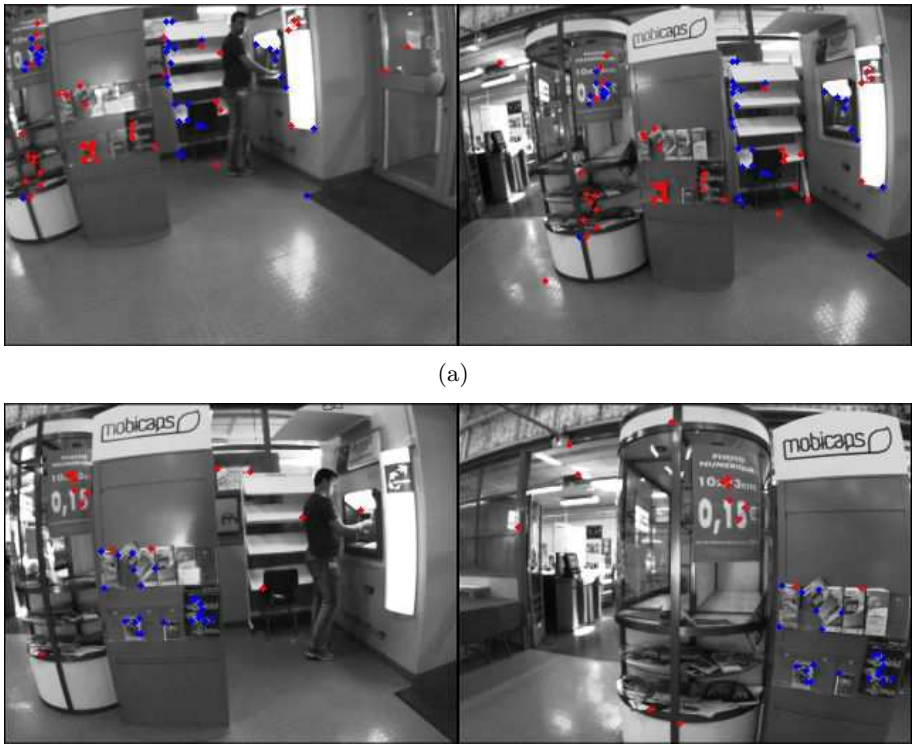
3.6.1 Première approche

Une première idée est de classer les points cohérents comme des points appartenant à la structure statique. Cependant lorsque les points de vue des images similaires sont trop éloignés ou lorsqu'une grande partie de la scène a bougé, il est difficile d'avoir une estimation précise de la pose et les points cohérents n'appartiennent pas forcément à la structure statique. La figure 3.8 illustre ce problème. Les points bleus sont les points respectant le calcul de pose. Les points rouges sont les points incohérents. Le panneau « Mobicaps » a effectivement bougé. Dans le cas présenté figure 3.8 (a), il est difficile de séparer les *inliers* des *outliers* de manière précise. Les points de vue sont trop éloignés. Certains points appartenant effectivement à la structure statique apparaissent incohérents. Dans le cas 3.8 (b), le modèle dominant trouvé est le modèle correspondant à l'objet mobile. Les points de l'objet sont des points cohérents et sont classés par erreur comme des points appartenant à la structure statique.

3.6.2 Filtrage temporel

Pour filtrer ces erreurs, nous devons prendre en compte les résultats obtenus sur l'ensemble de la séquence. Nous partons du principe que la structure statique est globalement dominante : elle possède le plus grand nombre de points 3D comparativement aux objets déplacés. Nous utilisons un système de vote pour enregistrer les résultats de l'ensemble des images clefs. A chaque fois qu'une vérification géométrique 2D/3D est exécutée, si un nombre minimum de points vérifient le calcul de pose, le statut de chaque point 3D observé par la caméra est mis à jour. A chaque itération et pour chaque point 3D, nous actualisons le nombre de cas dans lesquels le point apparaît cohérent et le nombre de cas dans lesquels il apparaît incohérent. La valeur maximale de ces deux nombres donne le statut du point 3D.

En pratique, il n'est pas nécessaire d'avoir traité l'ensemble de la séquence pour caractériser les points statiques et les points incohérents. Pour chaque objet déplacé, le traitement de l'ensemble des caméras observant l'objet en question suffit. Pour détecter des objets en ligne, nous validons le statut d'un point 3D dès qu'il a été vu au moins 3 fois. Nous verrons dans les expériences que ce procédé n'est pas toujours optimal mais il permet de détecter des objets en ligne : il n'est pas nécessaire d'avoir les résultats de l'ensemble de la séquence.



(a)

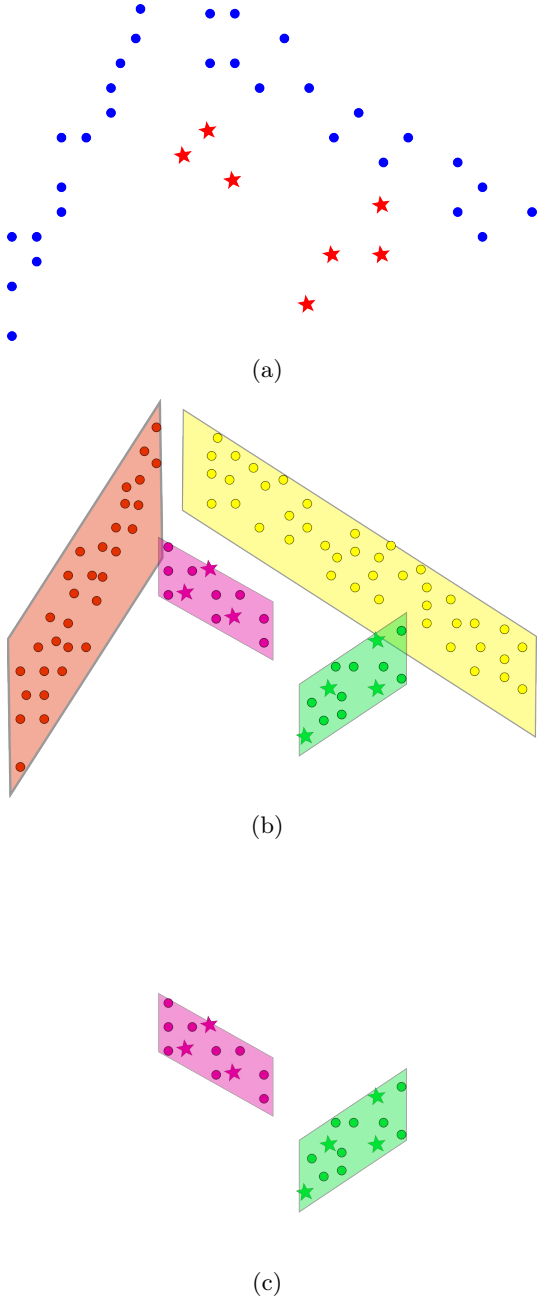
(b)

FIGURE 3.8 – Classification difficile des points cohérents et des points incohérents. Le panneau « Mobicaps » a effectivement bougé. (a) Calcul de pose imprécis : certains points appartenant effectivement à la structure statique apparaissent incohérents. (b) Calcul de pose reposant sur les points appartenant à l'objet déplacé : ces points sont considérés par erreur comme appartenant à la structure statique.

3.6.3 Classification

Une fois que les points de la structure statique et les points incohérents sont correctement séparés par la procédure de vote expliquée plus haut, nous devons caractériser les structures dynamiques correspondant aux objets déplacés. Pour cela nous appliquons la méthode illustrée sur le schéma de la figure 3.9. Les multiples structures sont détectées à partir des associations 2D/2D entre les deux vues similaires. Si une structure contient des points 2D associés à des points 3D incohérents, elle est caractérisée comme « objet ». Les résultats obtenus permettent d'enrichir la connaissance de la scène et des objets de deux manières :

- Dans un premier cas, l'objet est détecté pour la première fois et les points 3D ne sont pas associés à un objet dans la base. On enregistre alors le modèle d'apparence de l'objet, *i.e.* l'ensemble des descripteurs de la structure correspondante, et le modèle géométrique de l'objet constitué des points 3D associés.
- Dans le cas contraire, les points 3D sont déjà caractérisés comme appartenant à des objets. Le résultat permet alors de mettre à jour le modèle de l'objet précédemment détecté.



(a)

(b)

(c)

FIGURE 3.9 – **Caractérisation des structures dynamiques.** (a) Séparation des points de la structure statique (bleus) et des points incohérents (étoiles rouges). (b) Détection des multiples structures à partir des correspondances 2D/2D. (c) Objets détectés : structures possédant des points 2D associés à des points 3D incohérents.

Cette procédure respecte la définition donnée au début du chapitre (partie 3.1.2), à savoir, l'objet est défini comme un ensemble de points ayant un mouvement cohérent par rapport à la structure statique de la scène. De plus, elle permet d'enregistrer un modèle d'apparence et un modèle géométrique des objets détectés.

3.7 Résultats expérimentaux

Nous évaluons les performances de notre approche dans trois expériences différentes. La première expérience utilise une séquence vidéo tournée dans un environnement intérieur de grande taille. La trajectoire de la caméra forme une boucle : l'environnement est exploré deux fois et 6 objets ont bougé entre les deux passages. Les objets sont planaires. Ils ont été déplacés mais restent dans le même lieu. La deuxième expérience compare deux séquences prises dans un même bureau. Un certain nombre d'objets planaires ont bougés entre les deux passages. Ils ne restent pas forcément dans le même lieu. La troisième expérience met en œuvre la reconstruction d'une maquette de voiture articulée. La maquette est constituée d'un ensemble de sous-parties rigides pouvant bouger les unes par rapport aux autres. Cette expérience montre une application potentielle : la méthode peut être appliquée à l'apprentissage de la structure d'un objet déformable.

Les résultats obtenus dans les différentes expériences sont présentés dans les sections suivantes.

3.7.1 Expérience « Hall »

3.7.1.1 Jeu de données

TABLE 3.2 – Images représentatives de la séquence « Hall ». Les images sont ordonnées par ordre d'acquisition, suivant le déplacement de la caméra lors de l'expérience.

Nous évaluons les performances de notre algorithme sur un premier jeu de données constitué d'une séquence vidéo prise dans un environnement intérieur. Il s'agit d'une séquence de 2035 images. Des images représentatives de la séquence sont présentées sur le tableau 3.2. La caméra explore deux fois la même pièce et six objets planaires ont bougés entre les deux explorations.

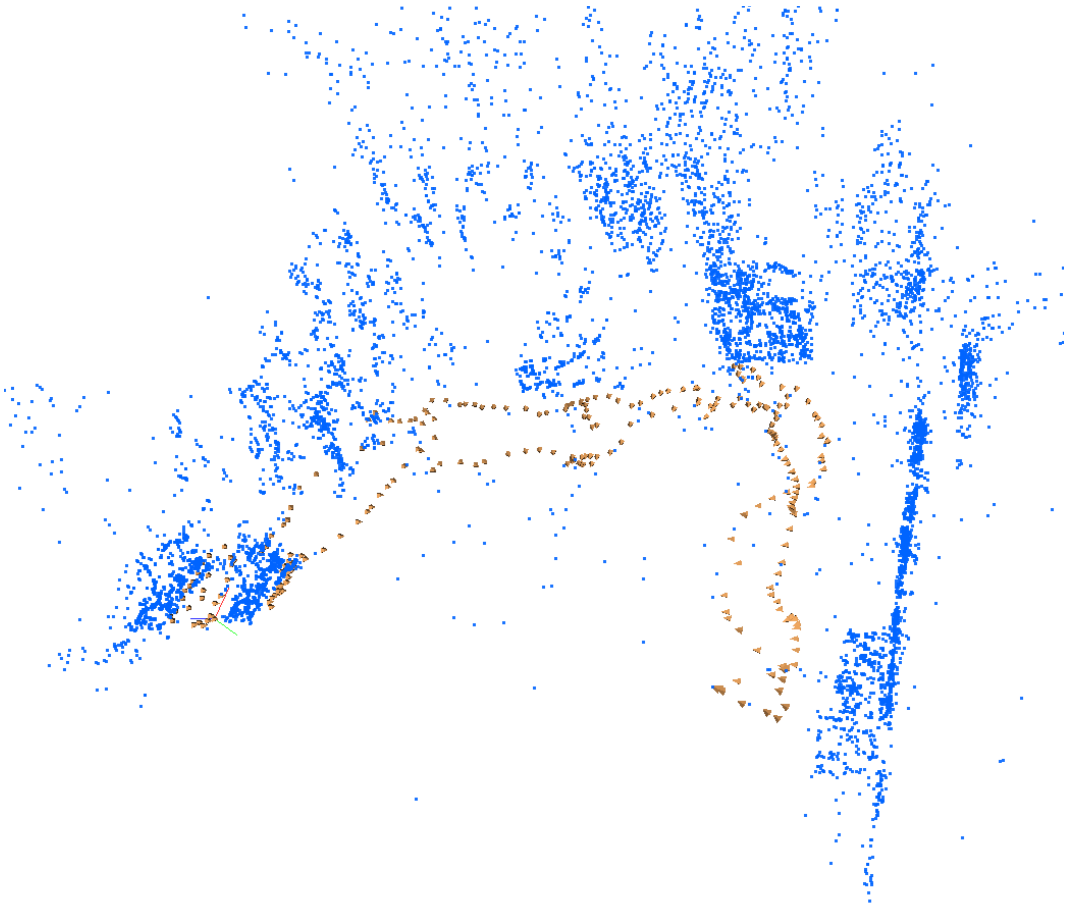


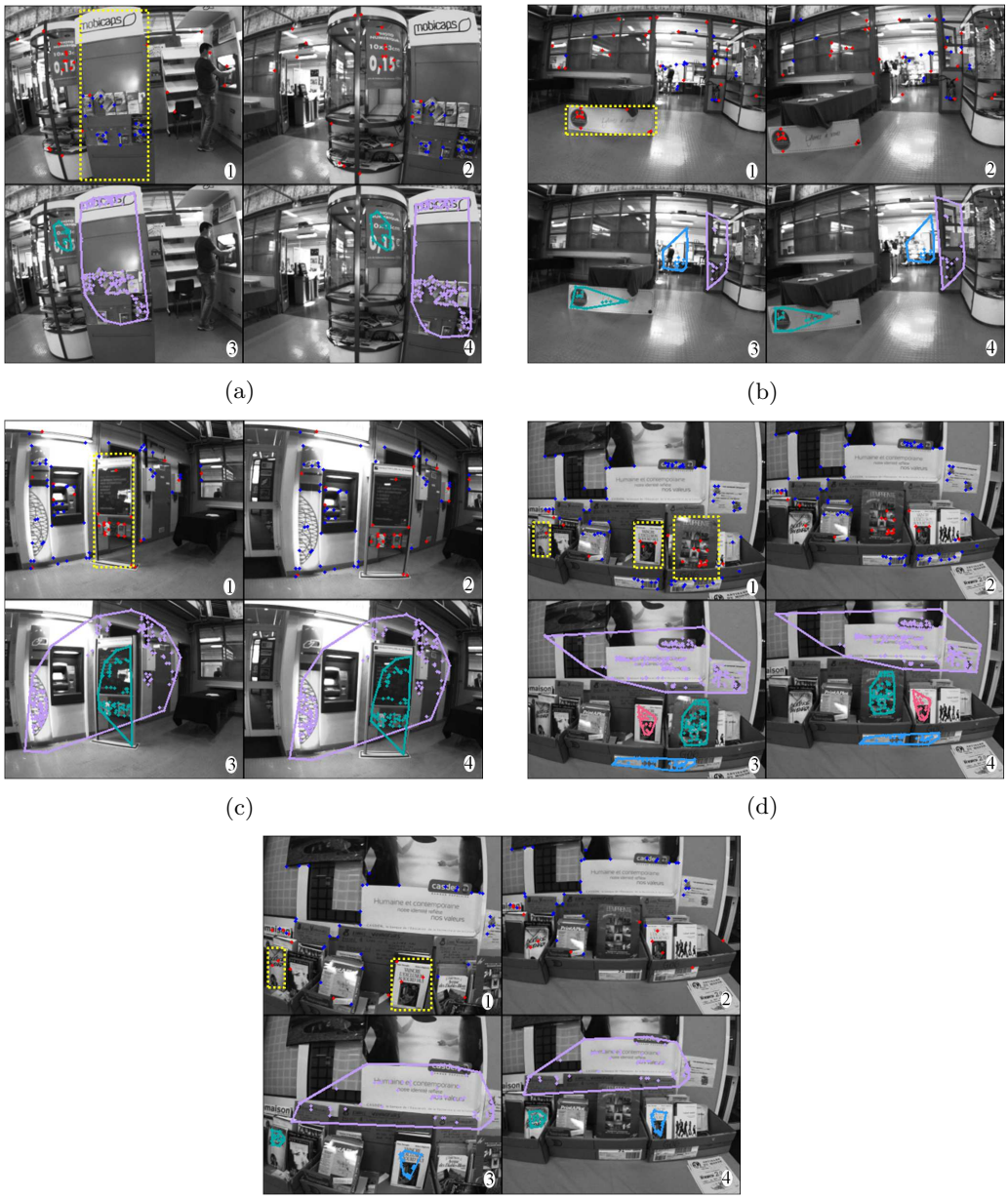
FIGURE 3.10 – Localisation et cartographie dans la pièce explorée lors de la première séquence. Les points 3D sont affichés en bleus, les positions des caméras associées aux images clefs sont représentées par des trièdres oranges.

3.7.1.2 Application de l'algorithme

La séquence d'images permet de reconstruire la pièce en 3D avec l'algorithme de SLAM métrique. 8164 points sont reconstruits avec l'ensemble de la séquence. Le nuage de points 3D et les positions des caméras associées aux images clefs sont affichés figure 3.10. Les premières images clefs sélectionnées sont enregistrées dans la carte topologique. Nous enregistrons les modèles d'apparence des 120 premières images clefs sur les 248 sélectionnées pour la reconstruction. A partir de la $121^{ième}$ image clef, on recherche pour chaque nouvelle image, l'image de la carte topologique la plus proche. C'est sur cette paire d'images similaires que nous détectons les multiples structures présentes. Les objets étant planaires, nous détectons les multiples homographies.

3.7.1.3 Extraction des objets : résultats

La figure 3.11 donne les résultats obtenus avec notre algorithme sur des cas de reconnaissance de lieu contenant des objets déplacés. Chaque figure représente deux



(a) (b)

(c) (d)

(e)

FIGURE 3.11 – **Expérience « Hall »** : Détection de structures sur cinq cas de reconnaissance de lieu. Vues 1 et 3 : image courante. Vues 2 et 4 : image similaire. Les points 3D sont affichés sur les vues 1 et 2 en bleu pour les points cohérents avec le calcul de pose, en rouge pour les points incohérents. Les vues 3 et 4 donnent les différentes homographies détectées dans les deux vues.

vues provenant d'un même lieu. La colonne de gauche (vues 1 et 3) représente l'image courante. L'image similaire trouvée dans la base de données est affichée sur la colonne de droite (vues 2 et 4). On observe les points de la carte 3D reprojetés sur les vues 1 (image courante) et 2 (image de la base de données). Les points cohérents avec le calcul de pose relative sont en bleu et les points incohérents en rouge. On indique aussi sur la vue 1 les objets qui ont effectivement bougé : les contours de ces objets sont surlignés en pointillés jaunes. Les vues 3 et 4 montrent les ensembles de points

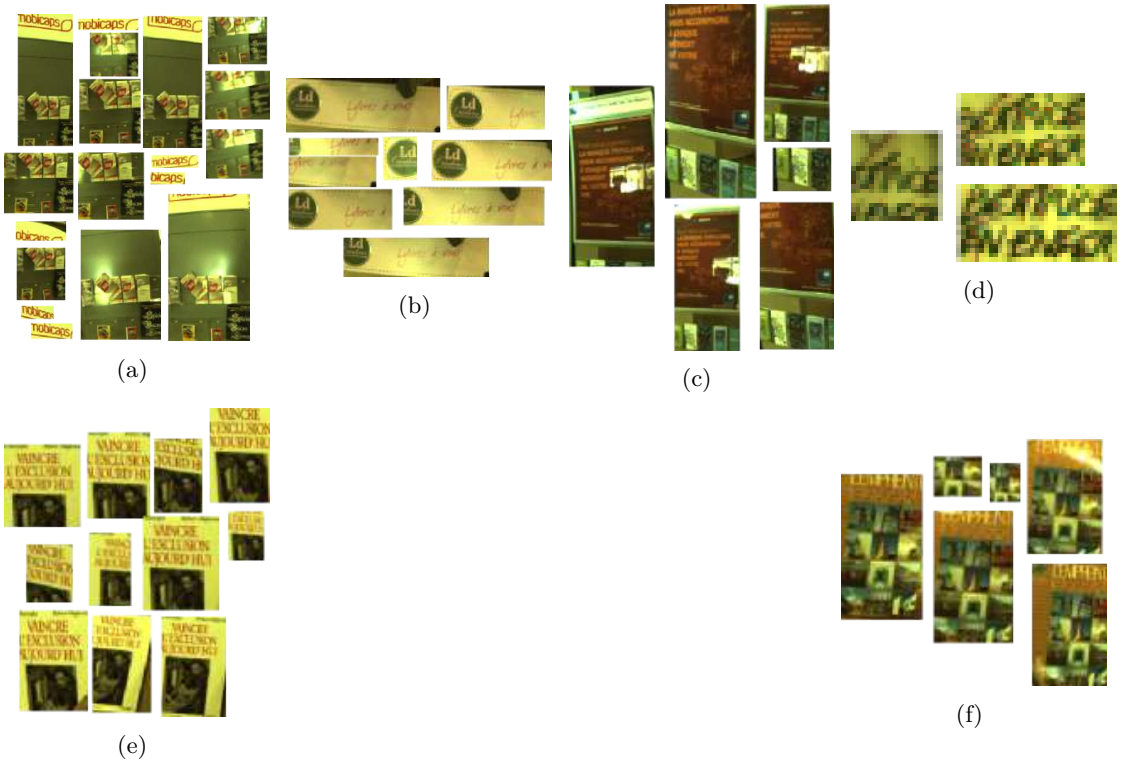


FIGURE 3.12 – **Expérience « Hall »** : ensemble des objets détectés par l'algorithme

FIGURE 3.13 – **Expérience « Hall »** : Nuage de points 3D de la structure statique (en bleu) et modèles géométriques des objets détectés (différentes couleurs).

appartenant à une même structure, détectés dans les deux vues similaires. Toutes les homographies détectées sont affichées. Les ensembles de points appartiennent donc à des structures statiques ou dynamiques. La comparaison de ce résultat avec la relocalisation permet d'extraire les objets. Dans les cas (b) et (c), un seul objet a bougé. La structure de l'objet est bien détectée (vues 3 et 4) et les points sont classés comme appartenant à un objet dynamique (vues 1 et 2). Dans le cas (a), un unique objet a été déplacé. Sa structure est bien détectée mais les points 3D correspondants sont catégorisés comme appartenant à la structure statique. Toutefois, les résultats précédents ont permis de classer ces points comme dynamiques grâce au système de vote décrit dans 3.6. De cette façon l'objet est correctement extrait. Dans le cas (c), trois objets sont détectés. Deux objets sont correctement détectés et catégorisés. La structure du $3^{ième}$ objet n'est pas détectée car le nombre de correspondances sur l'objet en question est relativement faible. Cet objet est tout de même extrait dans d'autres images. On le retrouve par exemple dans le cas (e).

Les résultats de notre algorithme sont moyennés sur 10 exécutions. On détecte en moyenne 53 ensembles de points 2D catégorisés en objets. Ces ensembles sont ensuite regroupés par objet en fonction des points 3D associés qu'ils contiennent. Deux ensembles sont réunis s'ils ont des points 3D associés en commun (voir section 3.6). Le modèle d'apparence de l'objet est alors formé par ces *clusters* et le modèle géométrique est donné par l'ensemble des points 3D associés. A chaque expérience, 100% des objets ayant effectivement bougé sont détectés. Le taux de fausse détection est de moins de 1% (moyenné sur 10 tests).

La figure 3.12 donne les résultats de l'expérience avec un aperçu des objets extraits par le traitement de l'ensemble de la séquence sur une exécution. Les vignettes des objets sont présentées figure 3.12. On note que le résultat (g) est une fausse détection. Ceci est dû aux imprécisions du calcul de pose relative. La séparation des points de la structure statique et des points incohérents n'est pas toujours précise et optimale et il arrive qu'un ensemble de points soit défini à tort comme un objet. Néanmoins, nous comptons un taux de fausse détection inférieur à 1%. De plus, cet objet n'est détecté qu'une seule fois, contrairement aux autres objets qui sont repérés dans plusieurs images. Le modèle d'apparence de chaque objet est constitué de l'ensemble des descripteurs des points 2D détectés dans ces vignettes. Les modèles géométriques sont représentés figure 3.13. Les points 3D de la structure statique sont affichés en bleu. Les ensembles de points 3D correspondants aux objets sont désignés par une couleur qui leur est propre.

3.7.2 Expérience « Bureaux »

Trois séquences ont été prises à l'intérieur d'une entreprise. Dans chacune des séquences, la caméra explore les deux mêmes bureaux. Le tableau 3.3 présente des images représentatives de l'environnement. Nous pouvons constaté l'importance de l'*aliasing* perceptuel dans ce jeu de données : beaucoup d'images sont des observations d'objets posés sur une table. Quinze objets ont été déplacés entre les deux premières prises de vue. Huit objets ont été déplacés entre la première prise et la troisième prise. Certains objets ont été bougés en restant dans le même lieu,

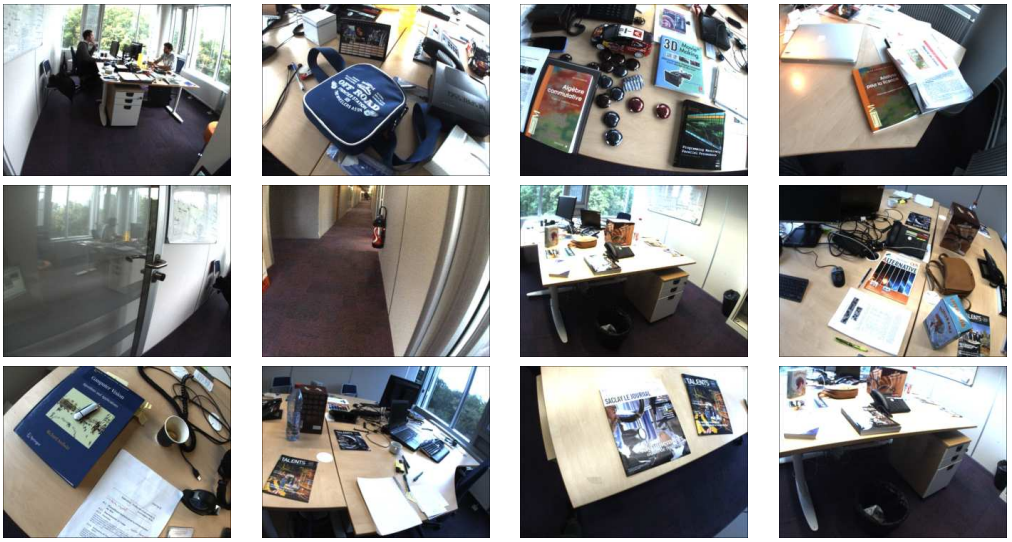


TABLE 3.3 – Images représentatives de la séquence « Bureaux ». Les images sont ordonnées par ordre d'acquisition, suivant le déplacement de la caméra lors de l'expérience.

d'autres sont déplacés dans l'autre pièce. La tableau 3.4 recense les informations des trois séquences, telle que le nombre d'images de la séquence (#Img), le nombre d'images clefs sélectionnées par l'algorithme de SLAM métrique (#Img clefs), le nombre d'objets bougés restant dans un même lieu (# Obj. bougé) et le nombre d'objets déplacés dans une autre pièce (# Obj. dép.). Nous utilisons la première

Nř Séquence	#Img	#Img clefs	# Obj. bougé	# Obj. dép.
1	2448	500	0	0
2	3561	476	9	6
3	1482	256	5	3

TABLE 3.4 – Descriptions des trois séquences « Bureaux »

séquence pour construire la carte topologique de l'environnement. Nous rappelons qu'elle est constituée de l'ensemble des lieux associés aux images clefs sélectionnées par l'algorithme de SLAM métrique. Deux lieux sont reliés topologiquement si des points observés dans les images clefs associées correspondent aux mêmes points 3D. Nous évaluons notre algorithme de découverte d'objet sur les deux autres séquences indépendamment. Les résultats respectifs des comparaisons de la séquence 2 et de la séquence 3 à la première séquence sont répertoriés dans le tableau 3.5, avec le nombre de détection d'objet correcte (# Dét. correcte) et le nombre de fausses détections (# Fausse dét.) et le nombre d'objets finalement extraits (# Obj. extrait). Dans chacune des expériences, notre algorithme parvient à détecter chaque objet au moins une fois, qu'ils soient restés dans le même lieu ou qu'ils aient été déplacés dans une autre pièce. Nous notons dans ces expériences un relativement fort taux de fausse détection de 25%. Les fausses détections surviennent toujours lorque la scène a fortement changé et qu'il est difficile de déterminer de façon précise la structure statique. Néanmoins les « faux objets » ne sont détectés qu'une seule fois, contrairement aux objets ayant effectivement bougé qui sont détectés plusieurs fois. La figure

Nř Séquence	#Img clefs	# Dét. correcte	# Fausse dét.	# Obj. extrait
2	476	153	44	15
3	256	62	8	8

TABLE 3.5 – Expérience « Bureaux » : Résultats de la découverte automatique des objets.

3.14 montre les structures statique et dynamiques détectées dans le premier bureau. La structure statique est représentée en bleu. Six objets sont représentés dans des couleurs différentes. Les vignettes des objets détectés sont représentées figure 3.6.

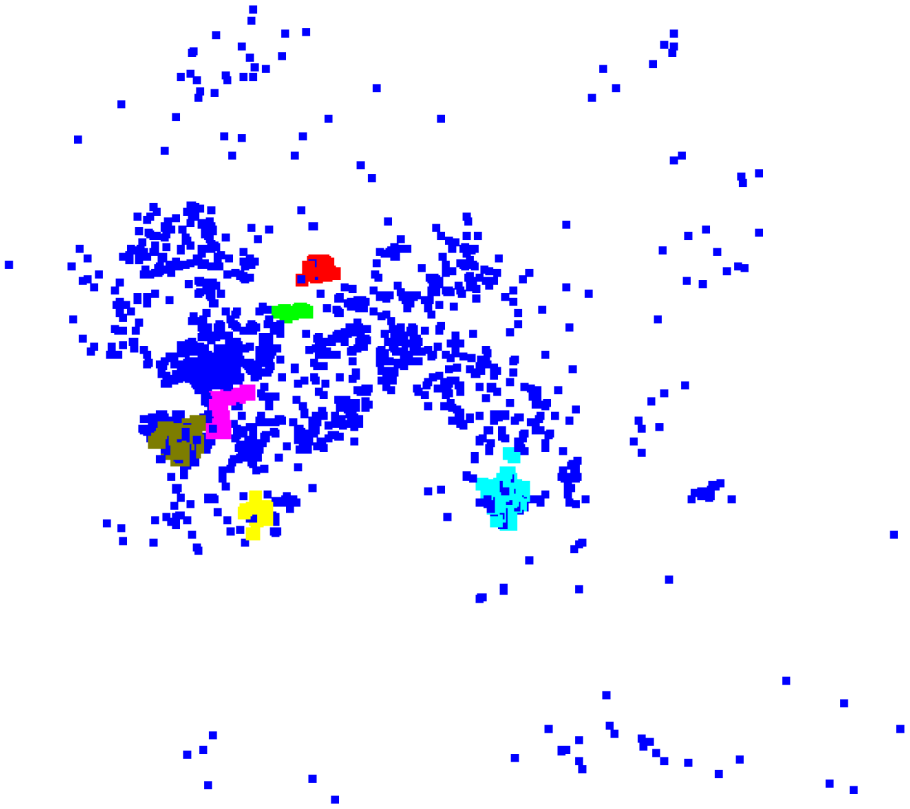


FIGURE 3.14 – **Expérience « Bureaux »** : Structure statique (bleu) et structure des objets détectés (différentes couleurs).

3.7.3 Expérience « Maquette »

Nous présentons dans cette partie les données utilisées lors de notre troisième expérience, ainsi que les résultats obtenus. La troisième expérience met en œuvre la reconstruction d'une maquette de voiture articulée. Deux images de cette maquette sont présentées figure 3.15. L'objet est constitué d'un ensemble de sous-parties rigides pouvant bouger les unes par rapport aux autres. Les portières et le capot de la voiture peuvent s'ouvrir. Au cours de la séquence, la caméra observe la maquette, portières et capot fermés, sous différents points de vue. Puis on ouvre une portière et le capot. La séquence contient 2500 images, elle permet de reconstruire 3570

TABLE 3.6 – Objets détectés avec la comparaison des séquences « Bureaux »

points avec la sélection de 127 images clefs. La première partie de la séquence sert de base d'apprentissage à la reconnaissance de lieu. Nous enregistrons les modèles d'apparence des 50 premières images clefs dans la carte topologique. A partir de la $51^{ième}$ image clef, nous recherchons pour chaque nouvelle image, l'image de la carte topologique la plus proche.

(a) (b)

FIGURE 3.15 – Expérience nř2. Images de la maquette de voiture. (a) La portière et le capot sont fermés. (b) La portière et la capot sont ouverts.

De la même façon que lors de la première expérience, nous recherchons les multiples structures présentes à partir des correspondances 2D/2D entre deux vues similaires. Ici, les parties articulées de la maquette sont presque planaires. Nous évaluons l'algorithme une première fois en détectant les multiples homographies et une deuxième fois en détectant les multiples matrices fondamentales. En effet, la portière et le capot ne sont pas des structures planes. Nous les approximons par des plans

dans une première expérience.

Dans une deuxième expérience, nous estimons les multiples matrices fondamentales. La figure 3.16 permet de comparer qualitativement les résultats obtenus dans les deux cas. Les figures de la colonne de gauche (a, c, e et g) présentent les multiples homographies détectées entre deux vues similaires. Les figures de la colonne de droite (b, d, f et h) sont les résultats obtenus avec la détection de multiples matrices fondamentales. Chaque paire d'images est un cas de reconnaissance de lieu. A gauche est affichée l'image courante, à droite l'image la plus proche dans la base de données. Les contours des parties de la maquette ayant effectivement bougé sont surlignées en pointillés jaunes. Les ensembles de points correspondants aux modèles trouvés par notre algorithme sont affichés en différentes couleurs. Nous constatons sur ces différents cas que la détection de multiples matrices fondamentales donne des résultats très imprécis. Sur la figure 3.16 (b), la structure de la portière ouverte n'est pas différenciée. Sur les figures 3.16 (d), (f), (h), les ensembles de points correspondants aux différentes structures contiennent beaucoup d'erreurs. Les différentes parties ne sont pas bien séparées.

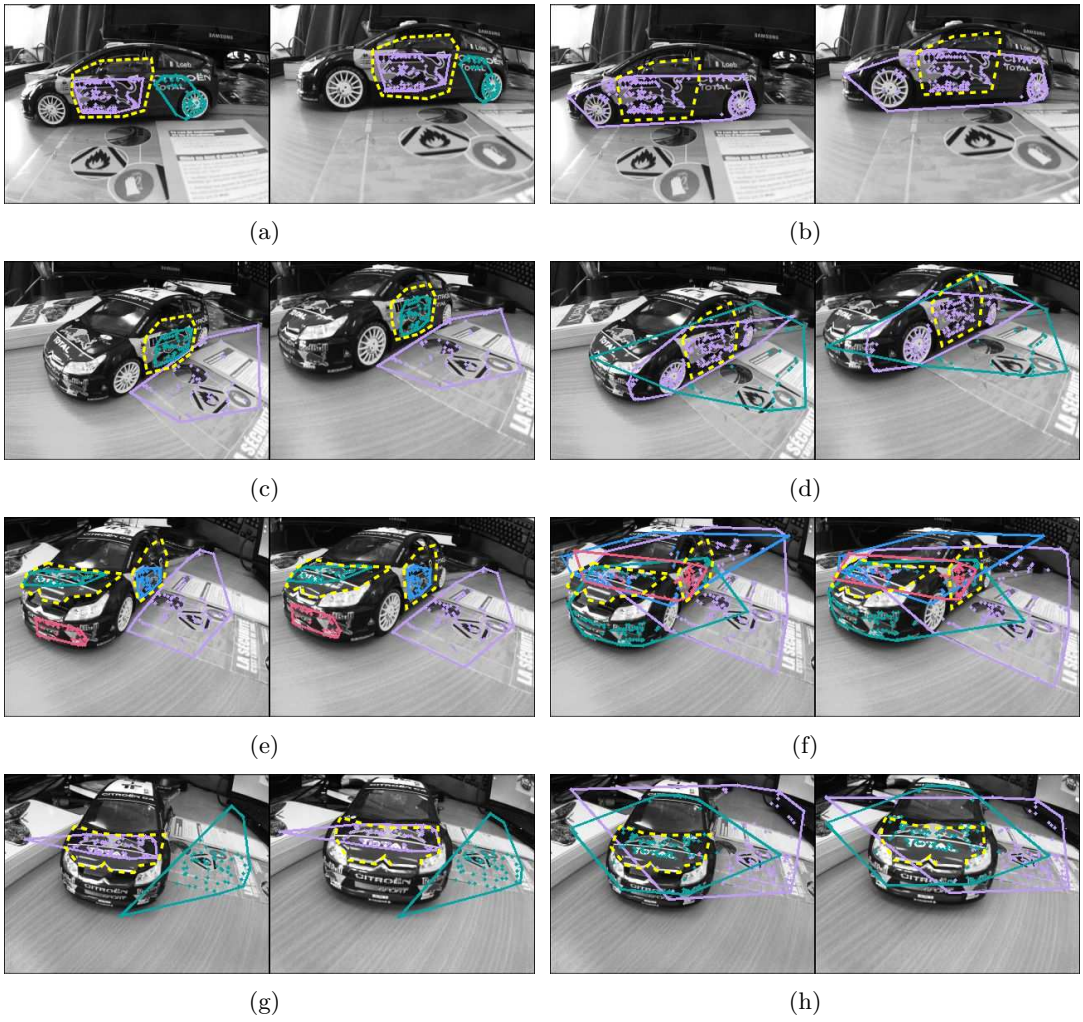
On note ici que la détection de multiples homographies obtient des meilleurs résultats. On compte à chaque exécution le nombre de détections ne correspondant pas à un objet déplacé. Nous estimons ainsi le taux de fausse détection. L'expérience est répétée 10 fois. En recherchant les multiples homographies, le traitement de l'ensemble de la séquence détecte en moyenne 15 structures avec un taux de fausse détection de 12%. En recherchant les multiples matrices fondamentales, le traitement de l'ensemble de la séquence détecte en moyenne 24 structures avec un taux de fausse détection supérieur à 70%.

3.8 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle formulation de la modélisation d'un environnement *a priori* inconnu en définissant explicitement la scène comme une structure statique et un ensemble d'objets dynamiques. Cette formulation offre la possibilité de modéliser la scène et son évolution au cours du temps. De plus, nous avons proposé une nouvelle approche, fonctionnelle et automatique, permettant de détecter les objets planaires déplacés dans une scène 3D et d'apprendre à la fois un modèle d'apparance et un modèle géométrique de ces objets. Notre méthode utilise les images provenant d'une unique caméra. Contrairement aux approches proposées par les auteurs de [Kim *et al.*, 2010] et [Castle *et al.*, 2010], l'algorithme proposé ne nécessite pas d'apprentissage hors ligne.

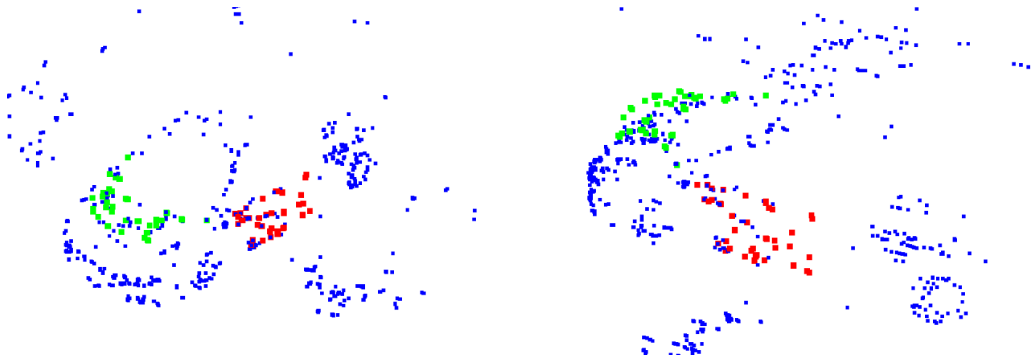
A ce jour, la méthode détecte des objets planaires ou des objets pouvant être approximer par des plans. Elle peut être étendue à la détection d'objets rigides quelconques en considérant qu'un objet est une concaténation de plusieurs plans.

Notre méthode combine les algorithmes de localisation par vision monoculaire et de détection de multiples structures entre deux vues. La détection de multiples structures est un problème complexe que nous étudions dans le chapitre 4. Nous



(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 3.16 – **Expérience nř2 :** Comparaison qualitative des résultats de détection de multiples structures. A gauche, les multiples homographies. A droite, les multiples matrices fondamentales.



(a) (b)

FIGURE 3.17 – **Expérience 2** : Nuage de points 3D de la structure statique (en bleu) et des modèles géométriques du capot et de la portière ouverts. Deux points de vue sont représentés sur cette figure.

présenterons dans ce chapitre, les approches proposées dans l'état de l'art ainsi qu'un nouvel algorithme de détection de multiples structures s'appuyant sur l'algorithme RANSAC.

Chapitre 4

Détection de multiples structures

Contenu du chapitre

Dans ce chapitre, nous étudions plus en détails le problème de détection de multiples structures entre deux vues. La résolution de ce problème est une étape essentielle à la méthode d'extraction automatique d'objet présentée dans le chapitre précédent. Nous présentons tout d'abord un état de l'art des méthodes existantes puis nous proposons notre algorithme reposant sur la procédure RANSAC que nous évaluons et comparons sur les problèmes d'estimation d'homographie et de géométrie épipolaire.

4.1 Etat de l'art

4.1.1 L'algorithme RANSAC

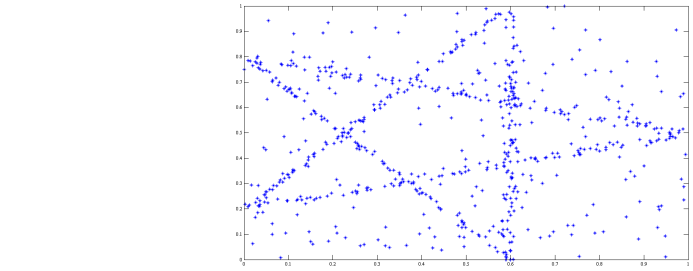
Notre méthode repose sur l'algorithme RANSAC présenté en détail dans le chapitre 2. RANSAC est une méthode itérative pour estimer les paramètres d'un modèle mathématique à partir d'un ensemble de données bruitées qui contient des valeurs aberrantes (*outliers*) [Fischler et Bolles, 1981]. Les données d'entrée de l'algorithme RANSAC sont un ensemble de données observées, un modèle paramétré qui peut être ajusté aux observations, et des paramètres d'intervalle de confiance. RANSAC atteint son objectif en sélectionnant itérativement un échantillon de taille minimale de façon aléatoire. L'échantillon est un sous-ensemble de données, la taille de l'échantillon est le nombre minimal de données nécessaires pour déterminer les paramètres du modèle. L'échantillon sélectionné est évalué de la façon suivante :

1. Un modèle hypothétique est ajusté aux données de l'échantillon.
2. Les autres données sont testées sur le modèle précédemment estimé. Si une donnée correspond au modèle estimé alors elle valide le modèle et est comptée comme *inlier*. On définit pour cela une mesure de distance au modèle et un seuil τ au dessus duquel les données ne sont pas considérées comme acceptables.
3. Le modèle est ré-estimé à partir de l'ensemble des *inliers*.

Cette procédure est répétée un nombre fixe de fois et le modèle retenu correspond au nombre maximal d'*inliers*. L'algorithme RANSAC est conçu pour estimer un unique modèle. Nous montrons dans ce chapitre comment adapter cette méthode à la détection de multiples modèles.

4.1.2 Définition du problème et vue d'ensemble des méthodes existantes

Des exemples de problème d'estimation de multiples structures sont illustrés figure 4.1. Il s'agit par exemple de déterminer les équations de droites présentes dans un ensemble de données 2D (figure 4.1 (a)). La figure 4.1 (b) présente une paire d'images stéréoscopiques. Les différentes structures planaires peuvent être segmentées en étudiant les correspondances de points entre les deux vues. De même, à partir de deux vues d'une même scène 3D prises à des instants différents, on cherche à segmenter les objets en mouvement. Nous définissons le problème de la manière



(a)

(b)

(c)

FIGURE 4.1 – **Illustration du problème de détection de multiples structures.** (a) Détection de droites à partir d'un ensemble de points 2D. (b) Détection de structures planaires à partir d'une paire d'images stéréoscopique. (c) Segmentation d'objets en mouvement (le cylindre, le pavé et la caméra ont bougé) à partir d'un flux vidéo.

suivante. Il s'agit de trouver les différentes instances d'un même modèle mathématique pour expliquer au mieux les observations. Le modèle mathématique peut être ainsi constitué des paramètres d'une droite, d'une homographie, d'une similitude ou d'une matrice fondamentale. Nous limitons notre étude à la recherche d'un seul type de modèle mathématique à la fois. Nous recherchons les diverses paramétrisations pour expliquer les observations. Pour simplifier, nous parlerons de modèle ou structure plutôt que de paramétrisation d'un modèle.

Dans ce mémoire, nous nous attachons notamment à la détection de multiples structures planaires et de multiples mouvements à partir de deux images similaires. La résolution de ce problème est une étape essentielle à notre méthode d'extraction

automatique d'objet présentée dans le chapitre 3. Le problème est complexe car nous devons simultanément regrouper les données selon leur appartenance à un même modèle et estimer les paramètres des modèles ainsi que leur nombre.

Chaque structure repose sur un ensemble de points consistants ou *inliers*. Dans le cas de présence de multiples structures, les données aberrantes (*outliers*) ont deux interprétations possibles. Ce sont soit des données erronées dues aux erreurs de mesure, soit des données consistantes avec d'autres paramètrisations qu'on appelle *pseudo-outliers*. Nous présentons dans la suite de cette section une vue d'ensemble des méthodes existantes de détection de multiples structures.

4.1.2.1 Adaptation de l'algorithme RANSAC

L'algorithme RANSAC, présenté dans le chapitre 2, partie 2.3, a été conçu pour estimer un unique modèle. Il n'est pas adapté à l'extraction de multiples structures. Une première idée est d'appliquer séquentiellement plusieurs procédures RANSAC successives en retirant du jeu de données les *inliers* de chaque nouveau modèle détecté. En pratique, lorsque plusieurs modèles sont présents, le nombre d'itérations nécessaires et le seuil séparant les *inliers* des *outliers* sont des paramètres difficiles à fixer. Cette approche n'est donc pas fonctionnelle.

Les auteurs de [Zuliani *et al.*, 2005] proposent une solution avec l'algorithme MultiRANSAC (désigné par 1-*MultiRANSAC* dans la suite du chapitre). Le nombre de modèle M étant connu, la procédure estime simultanément les paramètres des M modèles. A chaque itération, M échantillons de taille minimale sont formés. Les hypothèses sont générées et on obtient M groupes de points *inliers* correspondants. Ces groupes sont ensuite fusionnés avec ceux obtenus à l'itération précédente en gardant M groupes disjoints de cardinalité maximum. L'approche est effective mais nécessite la connaissance du nombre de modèle.

4.1.2.2 Approche par partionnement de données

De nombreuses méthodes répondent au problème de détection de multiples modèles en recherchant des amas dans un espace de projection. Le problème est alors formulé comme un problème de partitionnement de données (*data clustering*).

La transformée de Hough aléatoire (*Randomized Hough Transform*) est une méthode d'estimation robuste couramment utilisée en vision par ordinateur. La méthode sélectionne aléatoirement des échantillons de taille minimale et créé un histogramme dans l'espace paramétrique par un système de vote. Le modèle recherché correspond à un maximum dans l'histogramme. Les auteurs de [Subbarao et Meer, 2006] présentent une généralisation de la transformée de Hough adaptée à la présence de multiples structures. Les paramètres d'hypothèses générées aléatoirement et correspondant à la même structure sont proches et forment un amas dans l'espace des paramètres. La méthode projette les données dans l'espace des paramètres et recherche les modes de la distribution avec un algorithme mean-shift. Les modes

trouvés correspondent aux structures présentes. Cette méthode est désignée par 2-*Mean Shift* dans la suite du mémoire.

Une autre méthode, proposée par les auteurs de [Zhang et Kosecka, 2006] (désignée par 3-*Analyse des histogrammes de résidus* dans la suite du mémoire), analyse la distribution des résidus de chaque donnée par rapport aux hypothèses générées. Le résidu ou l'erreur résiduelle est la distance d'une observation au modèle calculé. Par exemple, pour une homographie ou une matrice fondamentale, elle peut être estimée avec la distance de Sampson présentée dans l'annexe A. Les modes de la distribution des résidus sont calculés par une méthode non paramétrique : le nombre de modèles présents et les paramètres de chaque modèle sont trouvés automatiquement.

Ces différentes méthodes se définissent comme étant entièrement automatiques. Néanmoins l'utilisateur doit indiquer le nombre d'hypothèses à générer initialement pour projeter les données d'entrée dans l'espace voulu.

Par ailleurs, les auteurs de [Toldo et Fusiello, 2008] ont montré expérimentalement que ces approches ne sont pas robustes aux *outliers*. Elles sont très perturbées par une forte proportion de données aberrantes, notamment lorsque le nombre de structures présentes augmente, ainsi que par une distribution inégale de points par structure. En effet, lorsque une structure est significative, *i.e* elle repose sur une grande partie des données, les hypothèses initiales ont plus de chance d'être générées à partir d'échantillons constitués uniquement de données consistantes. Dans ce cas, le pic correspondant est facile à détecter dans la distribution. Il est cependant très difficile de localiser ce pic pour une structure reposant sur peu de données.

Pour pallier ce problème, les auteurs de [Chin *et al.*, 2009] proposent d'utiliser une méthode d'apprentissage statistique qui reste effective avec un fort pourcentage d'*outliers* (4-*Clustering spectral* dans la suite du mémoire). N hypothèses initiales sont générées. Le nombre N doit être arbitrairement fixé pour assurer qu'un certain nombre d'échantillons soient consistants, c'est-à-dire constitués uniquement d'*inliers*. On obtient pour chaque point l'histogramme D des distances aux hypothèses générées en calculant les résidus. Les histogrammes $\{D_1, ..., D_n\}$ sont représentés par une matrice carrée définie par $K_{ij} = k(D_i, D_j)$, avec k est un noyau de Mercer (*i.e.* un noyau défini positif et continu) adapté. Cette astuce permet de projeter les données dans un espace de dimension plus élevée pour les segmenter plus facilement. La décomposition en valeurs singulières de la matrice carrée K permet de détecter et rejeter les données aberrantes. Ensuite, les différentes structures résultent de l'analyse en composantes principales de la matrice. La méthode est fonctionnelle mais elle n'est pas appropriée à un traitement en temps réel. De la même façon que pour les méthodes précédemment citées, l'utilisateur doit fixer le nombre N d'hypothèses initiales.

4.1.2.3 J-Linkage

Récemment, les auteurs de [Toldo et Fusiello, 2008] ont proposé une méthode effective d'estimation de multiples structures. Ils définissent un nouvel espace concep-

tuel de représentation des données de la manière suivante :

- N hypothèses sont générées au départ,
- chaque point est ensuite représenté dans l'espace conceptuel par une fonction caractéristique de l'ensemble des hypothèses préférées par ce point, à savoir les hypothèses pour lesquelles l'erreur résiduelle est inférieure à un certain seuil.

Le nombre N d'hypothèses à générer et le seuil permettant de classer une hypothèse dans l'espace conceptuel sont des paramètres arbitrairement fixés. Les modèles résultats sont extraits en regroupant les points dans l'espace conceptuel par un algorithme de partitionnement appelé *J-linkage*. Dans l'espace conceptuel, la distance entre deux points p_a et p_b est définie par la distance de Jaccard (4.1) entre les ensembles d'hypothèses préférées respectifs A et B :

$$d(p_a, p_b) = Jaccard(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{4.1}$$

La méthode est effective et robuste aux *outliers*. Elle est cependant moins précise lorsque les données sont bruitées et que le nombre de modèles augmente. Elle nécessite surtout un échantillonnage adapté pour réduire le nombre d'hypothèses initiales N et former des échantillons consistants reposant sur les structures effectivement présentes.

4.1.2.4 Minimisation d'énergie

Les auteurs de [Isack et Boykov, 2012] ont proposé récemment de résoudre le problème de détection de multiples structures par des algorithme de minimisation d'énergie. L'algorithme RANSAC peut en effet être interprété comme un problème de minimisation d'energie. On peut considérer que le RANSAC trouve les paramètres L du modèle correspondant au plus grand nombre d'*inliers* ; les *inliers* étant les points dont la distance au modèle est inférieure à un seuil τ. Il s'agit donc de minimiser l'énergie :

$$E(L) = \textstyle\sum_p \|p - L\|$$

avec

$$\|p - L\| = \begin{cases} 0 & \text{si } dist(p, L) < \tau \\ 1 & \text{sinon} \end{cases}$$

et $dist(p, L)$ une mesure de la distance entre le point p et le modèle de paramétrisation L. La formulation MSAC introduite dans [Torr et Zisserman, 2000] permet d'avoir une estimation plus précise du modèle en utilisant la norme :

$$\|p - L\| = \begin{cases} dist(p, L) & \text{si } dist(p, L) < \tau \\ T & \text{sinon} \end{cases}$$

Lorsque plusieurs modèles sont présents, l'énergie doit être minimisée sur l'ensemble des labels $\mathbf{L} = \{L_p\}$, L_p étant le modèle assigné au point p. La minimisation de l'énergie

$$E(L) = \sum_p \|p - L_p\| \tag{4.2}$$

n'est pas adaptée car on observe un phénomène de surajustement (*overfitting*). La solution donne alors autant de modèles que d'hypothèses générées.

Les auteurs de [Isack et Boykov, 2012] ont proposé une régularisation de l'énergie 4.2 permettant de limiter le nombre total de modèles trouvés et d'encourager une cohérence spatiale (les données spatialement corrélées doivent appartenir à la même structure). N hypothèses initiales sont générés par échantillonnage aléatoire et les auteurs appliquent l'algorithme d'α-*expansion* introduit dans [Boykov *et al.*, 2001] pour minimiser l'énergie régularisée. L'article montre de bons résultats mais la méthode ne nous semble pas appropriée. En effet, en pratique l'utilisateur doit fixer trois paramètres : le choix des valeurs impacte beaucoup les résultats et peut dépendre du problème et des données. On désigne cette méthode par 6-*Minimisation de l'énergie* dans la suite du chapitre.

4.1.3 Limites actuelles

4.1.3.1 Paramètres à fixer

Les méthodes que nous avons citées dans l'état de l'art requièrent un certain nombre de paramètres que l'utilisateur doit fixer. Les différents paramètres sont listés dans le tableau 4.1. Les méthodes de partionnement 2-*Mean Shift* [Subbarao et Meer, 2006], 3-Analyse des histogrammes de résidus [Zhang et Kosecka, 2006] et 4-*Clustering spectral* [Chin *et al.*, 2009] ne fixent pas de seuil pour séparer les données consistantes des données aberrantes. Néanmoins, lors du partitionnement, il faut indiquer un seuil pour séparer les données dans des groupes ou *clusters*. Ce seuil s'apparente à la frontière *inliers*/*outliers* utilisée par les autres méthodes. Les auteurs de [Isack et Boykov, 2012] proposent une méthode à première vue complètement automatique. Cependant les coefficients de pondération utilisés pour l'énergie regularisée reviennent à fixer un seuil séparant les *inliers* des *outliers* et à définir la taille minimale qu'une structure doit avoir. D'autre part, nous avons observé que les résultats de la méthode variaient beaucoup selon la valeur des coefficients et selon les données de test.

	Méthodes					
Paramètres	1	2	3	4	5	6
Bruit de mesure	×				×	
Nombre de structures	×					
Taille moyenne d'une structure		×			×	×
Seuil séparant les *inliers* des *outliers*	×				×	×
Nombre d'hypothèses à générer	×	×	×	×	×	×

TABLE 4.1 – Paramètres à fixer par l'utilisateur pour chaque méthode. 1-MultiRANSAC [Zuliani *et al.*, 2005], 2-*Mean-Shift* [Subbarao et Meer, 2006], 3-Analyse des histogrammes de résidus [Zhang et Kosecka, 2006], 4-*Clustering* spectral [Chin *et al.*, 2009], 5-JLinkage [Toldo et Fusiello, 2008], 6-Minimisation de l'énergie [Isack et Boykov, 2012]

4.1.3.2 Nombre d'hypothèses à générer

Toutes les méthodes précédemment citées commencent par la génération d'un ensemble de modèles candidats (hypothèses). La génération d'une hypothèse revient à effectuer une itération de l'algorithme RANSAC. Quelle que soit la méthode, pour obtenir des résultats probants, il faut s'assurer qu'un certain nombre d'hypothèses soient **pertinentes**, *i.e* qu'elle soient calculées à partir d'échantillons non contaminés. Le nombre théorique d'itérations nécessaires pour trouver une solution s'accroît très vite avec le taux d'outliers $1 - l$ et la compléxité du modèle. On note m le nombre minimal de données nécessaires pour calculer les paramètres d'un modèle. m vaut 4 pour une homographie et 7 pour une matrice fondamentale. Le tableau 4.2 rappelle le nombre d'itérations nécessaires pour sélectionner au moins un échantillon non contaminé.

m \ l	80%	60%	40%	20%	15%	10%	5%
4	9	33	178	2900	$9.1 \cdot 10^3$	$4.6 \cdot 10^4$	$7.4 \cdot 10^5$
7	20	162	2800	$3.6 \cdot 10^4$	$2.7 \cdot 10^6$	$4.6 \cdot 10^7$	$5.9 \cdot 10^9$

TABLE 4.2 – Nombre d'échantillons de m données qu'il faut générer pour former au moins un échantillon sans données erronées avec une probabilité de 99% en fonction du taux de données correctes l.

Dans le cas où plusieurs instances du modèle recherché co-existent, chaque structure repose sur une partie des données. Il est alors plus difficile et peu probable de générer un échantillon non contaminé pour un modèle particulier. Par exemple, si trois modèles coexistent dans les données, le pourcentage de points appartenant à chaque structure est de 33%. Pour un échantillon de 4 points, la probabilité de générer un échantillon non contaminé pour un modèle vaut $0.33^4 = 0.012$. En supposant que les instances contiennent le même nombre de données, le tableau 4.3 donne la probabilité de former un échantillon non contaminé en faisant varier le nombre d'instances. On suppose ici qu'il n'y a pas de données erronées. Avec des données réelles, la probabilité est encore réduite. Dans ces conditions, l'échantillonnage aléatoire devient trop coûteux et il devient indispensable d'utiliser une méthode d'échantillonnage adaptée. Générer des hypothèses pertinentes plus rapidement permet d'accélérer l'ensemble des méthodes et d'améliorer la précision des solutions.

m \ **Insts.**	**1**	**2**	**3**	**4**
4	100%	6.25%	1.2%	0.39%
7	100%	0.78%	0.04%	0.006%

TABLE 4.3 – Probabilité de former un échantillon constitué de points appartenant à la même structure dans le cas où plusieurs instances coexistent. On suppose qu'il n'y a pas de données erronées et que chaque structure est composée du même nombre de données.

4.1.4 Conclusion

Nous avons présenté dans cette section les méthodes de la littérature répondant au problème de détection de multiples structures. Les différentes approches ne fonctionnent pas de manière complètement autonome : l'utilisateur doit fixer un certain nombre de paramètres. D'autre part, chacune des méthodes repose sur la génération d'hypothèses et peut être améliorée et accélérée en utilisant un échantillonnage adapté.

Nous présentons un algorithme complet de détection de multiples structures dans la partie 4.2. Il s'appuie sur l'algorithme RANSAC. L'utilisateur doit fixer la valeur du seuil séparant les *inliers* des *outliers* et le nombre d'hypothèses à générer. Néanmoins il ne nécessite pas d'information supplémentaire comme le nombre de modèles présents ou la taille minimale d'une structure. Le nombre d'hypothèses à générer peut être fixé pour respecter des contraintes de temps de calcul car nous évaluons les hypothèses pour ne garder que les modèles pertinents (voir partie 4.5.1). L'algorithme proposé est comparé à la méthode J-Linkage [Toldo et Fusiello, 2008] et à la méthode de minimisation de [Isack et Boykov, 2012], deux méthodes qui ont montré des résultats probants.

Par ailleurs, nous avons aussi travaillé à l'élaboration de nouvelles méthodes d'échantillonnage pour accélerer les méthodes existantes. Nous présentons ces procédures dans la partie 4.3. Ces méthodes sont évaluées et comparées aux méthodes existantes sur des données synthétiques et réelles.

4.2 Algorithme de détection de multiples structures

Nous présentons dans cette section un nouvel algorithme de détection de multiples structures. Dans nos travaux, nous avons étudié la détection de plans et la détection de mouvements à partir d'une paire d'images provenant d'un même lieu. Pour la détection de plans, il s'agit de trouver les multiples homographies présentes à partir des correspondances de points entre les deux images. Pour la détection de mouvements, nous recherchons les multiples matrices fondamentales. Nous proposons un algorithme rapide qui limite le nombre de paramètres à fixer.

Notre approche combine deux méthodes complémentaires :
- *BetaSAC* (section 4.3.4) créé des échantillons menant à des hypothèses pertinentes.
- une étape d'optimisation (LOSAC) qui s'appuie sur les travaux de [Chum *et al.*, 2003] permet d'améliorer la qualité de la solution lorsqu'un nouveau modèle est trouvé.

L'approche est résumée dans l'algorithme 2 et illustrée par la figure 4.6.

4.2.1 Préambule

4.2.1.1 Définitions et rappels

Nous rappelons ici les définitions nécessaires à la compréhension du chapitre. Un *échantillon* est un sous-ensemble de m correspondances. m étant inférieur au nombre total de correspondances. La taille minimale de l'échantillon est le nombre minimal de correspondances nécessaire pour déterminer les paramètres d'un modèle. Une *hypothèse* est générée à partir d'un échantillon de taille minimale. Le *support* géométrique d'une hypothèse est l'ensemble des correspondances consistantes avec la paramétrisation de l'hypothèse, *i.e* en accord avec le modèle ajusté aux données de l'échantillon. Une *structure* est un ensemble de points correspondant effectivement à un plan ou à un objet en mouvement dans la scène.

Un échantillon consistant est un *échantillon* constitué uniquement de points appartenant à la même structure. L'hypothèse générée à partir d'un tel échantillon est une hypothèse *pertinente*.

4.2.1.2 Paramètres de la méthode

L'utilisateur doit fixer arbitrairement le seuil θ qui définit la frontière entre les *inliers* et les *outliers* d'un modèle. Dans les expériences, ce seuil est de $1,5$ pixels.

L'utilisateur doit aussi définir un critère d'arrêt, il peut le faire de différentes manières :
- le nombre d'hypothèses à générer peut être choisi en fonction des contraintes de temps de calcul,
- l'algorithme est arrêté lorsqu'un certain pourcentage d'*inliers* ont été trouvés,
- l'algorithme est arrêté lorsqu'un certain nombre de structures ont été détectées.

On détermine avec le critère choisi, le nombre d'itération de l'algorithme T. Dans les expériences réalisées, nous avons fixé le nombre d'hypothèses maximal générées.

4.2.1.3 Notations

On note C l'ensemble des correspondances entre les deux images similaires. Ce sont les données d'entrée de notre algorithme. On note S un sous-ensemble de C. On note S_l le sous-ensemble de C constitué de l'ensemble des points consistants avec le $l^{ième}$ modèle trouvé. L'algorithme trouve la collection $\{S\}$ d'ensembles de points appartenant aux différentes structures de la scène.

4.2.2 Vue d'ensemble de la méthode proposée

L'algorithme 2 résume les étapes de notre méthode.

Algorithm 2 Détection de multiples structures

Require: C, θ, T

$\{S\} \leftarrow \emptyset$, $t \leftarrow 0$
while $t \leq T$ **do**
  $t \leftarrow t + 1$
  for $l = 1$ to $L_{t-1} + 1$ **do**
    $S_{Hyp} \leftarrow$ BetaSAC(D)
    $\{S^t\}_{Hyp} \leftarrow$ ajoute(S_{Hyp})
    $D \leftarrow D \backslash S_l^{t-1}$
  end for
  $\{S^t\}_{Hyp} \leftarrow$ optimise($\{S^t\}_{Hyp}$, θ)
  for S in $\{S^t\}_{Hyp}$ **do**
    if $variance\_du\_bruit(S) > \theta$ **then**
      $\{S^t\}_{Hyp} \leftarrow \{S^t\}_{Hyp} \backslash S$
    end if
  end for
  $\{S^t\} \leftarrow$ fusionne($\{S^{t-1}\} \cup \{S^t\}_{Hyp}$)
end while
return $\{S\}$

Supposons qu'à l'itération $t-1$, l'algorithme ait déjà trouvé L_{t-1} structure différentes. Nous avons donc à $t-1$, la collection $\left\{S_l^{t-1}, l = 1, ..., L_{t-1}\right\}$ d'ensembles de points appartenant aux L_{t-1} structures détectées. A l'itération t, nous générons une nouvelle collection d'hypothèses $\{S^t\}_{Hyp}$. Pour cela, un premier échantillon de taille minimale est sélectionné suivant la stratégie BetaSAC (section 4.3.4). Il résulte de cette hypothèse un ensemble d'*inliers* S_{Hyp} appartenant à une structure hypothétique. Nous répétons cette procédure L_{t-1} fois, en retirant à chaque fois l'ensemble S_l^{t-1} d'*inliers* trouvés à l'itération précédente. Cette opération, inspirée du RANSAC séquentiel permet d'augmenter les chances de détecter les structures les plus petites. Il est difficile de former des échantillons consistants avec une structure reposant sur très peu de points. Le fait de retirer du jeu de données les ensembles de points correspondant aux structures déjà détectées (qui sont généralement les plus grandes) permet d'augmenter les chances de former de tels échantillons.

Les hypothèses $\{S^t\}_{Hyp}$ sont ensuite optimisées suivant la méthode 4.5 présentée plus loin.

Nous évaluons les hypothèses optimisées en estimant la variance du bruit de mesure à partir des résidus des points (fonction *variance_du_bruit* dans l'algorithme). L'estimation de la variance est expliquée en détail dans la partie 4.5.1, elle permet de mesurer la qualité d'une détection. Nous rejetons les hypothèses pour lesquelles la variance est supérieure à θ.

La nouvelle collection $\{S^t\}$ est formée en combinant $\{S^{t-1}\}$ avec les ensembles de $\{S^t\}_{Hyp}$ non rejetés suivant la méthode décrite dans la section 4.5.2.

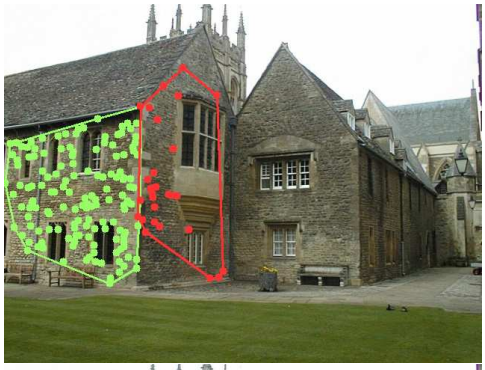


FIGURE 4.2 – Itération $t-1$: 2 plans trouvés.

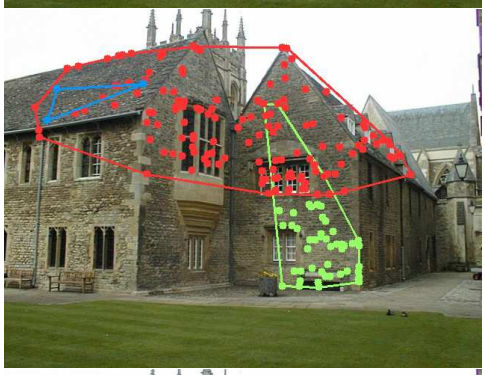


FIGURE 4.3 – Itération t : 3 hypothèses générées.

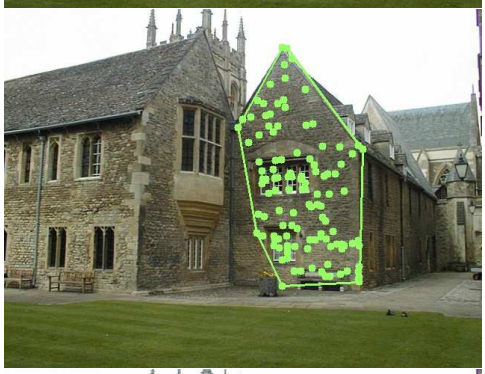


FIGURE 4.4 – Itération t : optimisation et évaluation, deux hypothèses ont été rejetées.

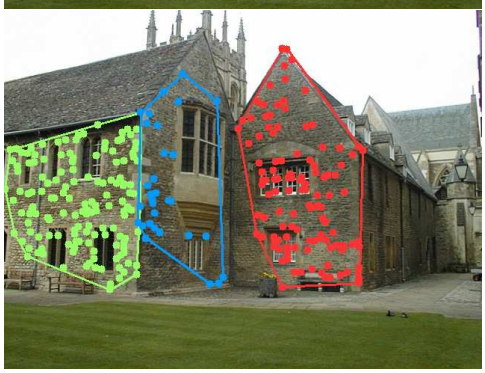


FIGURE 4.5 – Itération t : 3 plans trouvés après l'étape de fusion.

FIGURE 4.6 – Une itération de l'algorithme appliqué à la détection de multiples plans.

4.2.3 Illustration d'une itération de l'algorithme

Nous présentons ici une itération de l'algorithme appliqué à la détection de multiples homographies. Les données utilisées sont les deux images de la figure 4.1 (b). Les plans recherchés sont les différentes façades du bâtiment. Les ensembles de points

d'une même couleur vérifient la même homographie. La figure 4.2 montre les 2 plans trouvés à l'itération $t-1$. Sur la figure 4.3, les 3 groupes de points correspondent aux trois hypothèses générées à l'itération t : c'est la collection $\{S^t\}_{Hyp}$. La figure 4.4 présente les hypothèses retenues après les étapes d'optimisation et d'évaluation. Enfin, les résultats obtenus à l'itération t sont présentés figure 4.5.

4.3 Nouvelles méthodes d'échantillonnage

Nous présentons dans cette partie deux nouvelles méthodes d'échantillonnage. Un échantillonnage adapté est indispensable au bon fonctionnement de notre algorithme. Il permet aussi d'améliorer les méthodes existantes de détection de multiples structures entre deux vues.

4.3.1 Stratégies d'échantillonnage existantes

Les différents algorithmes présentés font le constat qu'une stratégie d'échantillonnage adaptée limite le nombre d'itérations nécessaires. D'après le paragraphe précédent, elle est indispensable pour l'estimation d'homographies et de matrices fondamentales. Pour ces deux problèmes, les modèles sont instanciés à partir d'un ensemble de correspondances de points de taille minimale (4 pour une homographie, 7 pour une matrice fondamentale) entre deux images similaires. Les solutions proposées utilisent en général la cohérence spatiale des points appartenant à la même structure qui sont généralement regroupés dans le plan de l'image.

Les auteurs de [Schindler et Suter, 2006] proposent de former des échantillons en utilisant des points situés dans une même sous-région de l'image. La figure 4.7 donne un exemple de sous-régions utilisées. L'image est divisée en trois lignes et trois colonnes qui se recouvrent partiellement. Les échantillons sont sélectionnés dans l'image entière, dans chaque colonne, chaque ligne et chacune des neuf régions définie par une intersection ligne-colonne. Les auteurs montrent que si une structure couvre $\approx 10\%$ de l'image entière, il y a au moins une région dans laquelle elle couvre $\approx 50\%$ de la zone. Pour l'estimation d'homographies, le nombre d'itérations nécessaires passe d'environ 46000 à 1200.

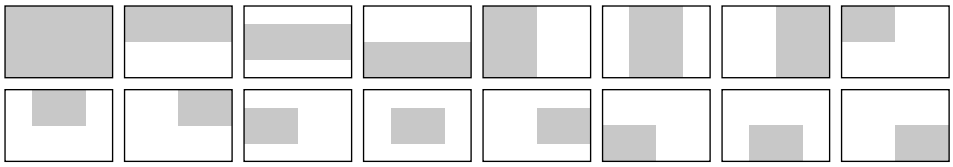


FIGURE 4.7 – Echantillonnage local utilisé par [Schindler et Suter, 2006] : les points sélectionnés pour former un échantillon proviennent de la même sous-région (partie grisée).

Les méthodes [Toldo et Fusiello, 2008], [Isack et Boykov, 2012] et [Chin *et al.*, 2009] utilisent un processus d'échantillonnage favorisant la sélection de points voi-

sins. Si un point x_i a déjà été sélectionné, alors la probabilité de sélectionner x_j est :

$$P(x_i|x_j) = \begin{cases} \frac{1}{Z} exp(-\frac{\|x_i - x_j\|^2}{\sigma^2}) & \text{si } x_i \neq x_j \\ 0 & \text{si } x_i = x_j \end{cases} \quad (4.3)$$

où σ est fixé par l'utilisateur et Z est un coefficient de normalisation. Ce paramètre représente la taille moyenne d'une structure présente. Cet échantillonnage comporte deux inconvénients. La sélection nécessite la connaissance des distances entre toutes les données, ce qui représente un long calcul (en $\mathcal{O}(N^2)$) et nécessite beaucoup de mémoire. De plus le paramètre σ dépend de la taille des structures présentes et doit être fixé par l'utilisateur. Cette méthode est désignée par le diminutif *Exp* dans les parties suivantes.

Récemment, les auteurs de [Chin *et al.*, 2012] ont proposé une nouvelle méthode d'échantillonnage baptisée *MultiGS*. L'algorithme utilise les résultats des itérations précédentes pour guider les itérations suivantes. Néanmoins, les performances de la méthode n'ont pas été évaluées dans un algorithme complet de détection de multiples structures.

4.3.2 Motivations

Les échantillons non contaminés sont les échantillons ne contenant aucune donnée erronée. Ce sont les échantillons que l'on souhaite générer le plus tôt possible. L'échantillonnage aléatoire classique n'utilise aucune information supplémentaire lors de la construction de l'échantillon. Il utilise une liste non ordonnée de points et chaque échantillon a alors la même chance d'être sélectionné. Les informations acquises lors des mesures dans l'image donnent pourtant des indications intéressantes. Elle permettent de regrouper des points appartenant potentiellement à la même structure avant tout autre calcul de vérification géométrique. L'idée d'utiliser des informations disponibles lors de l'échantillonnage n'est pas nouvelle.

Les auteurs de [Chum et Matas, 2005] favorisent la sélection de points ayant un fort score de mise en correspondance (PROSAC). Néanmoins, lorsque plusieurs instances coexistent, il n'y a aucune raison que ce score soit discriminant. Un échantillon de ce type contient des données correctes mais il a de fortes chances de contenir des données associées à différentes structures et mène à une hypothèse inutile.

Les auteurs de [Ni *et al.*, 2009] proposent avec l'algorithme GROUPSAC, de sélectionner des points appartenant à un même groupe. Ces groupes étant formés préalablement par une étape de segmentation basée sur la couleur et la texture. L'information est utile mais le temps de calcul additionnel est important.

Nous utilisons l'algorithme BetaSAC présenté dans le chapitre 2 en section 2.3.2. BetaSAC est une méthode d'échantillonnage conditionnel permettant de former des échantillons pertinents (*i.e* susceptibles de mener à une estimation précise de la transformation) plus rapidement qu'un échantillonnage aléatoire. L'algorithme permet d'exploiter n'importe quel type d'information lors de la construction d'un échantillon. Il suffit de définir une fonction de tri adaptée. De plus, le temps de calcul

additionnel est négligeable.

Nous montrons dans les prochaines sections comment adapter l'algorithme BetaSAC au problème de détection de multiples structures. Deux nouvelles stratégies sont proposées et évaluées. La première utilise l'information de proximité spatiale. La seconde utilise l'information tirée des tirages précédents qui s'avère utile pour guider et améliorer les tirages suivants. Ces stratégies sont évaluées et comparées sur plusieurs jeux de données synthétiques et réelles pour l'estimation de similitudes, d'homographies et de matrices fondamentales

4.3.3 Notations

Les données d'entrée sont un ensemble $C = \{C_1, ..., C_N\}$ de N correspondances de points détectés dans deux images. Les points détectés dans le plan de l'image sont représentés par le vecteur homogène $p = \begin{pmatrix} x & y & 1 \end{pmatrix}$. Le modèle recherché est déterminé à partir d'un sous-ensemble $s = \{s_i\}_{i=1}^{m}$ de m correspondances (m est la taille minimale pour déterminer le modèle). m vaut 2 pour une similitude, 4 pour une homographie et 7 pour une matrice fondamentale.

4.3.4 BetaSAC spatial : tri basé sur l'information spatiale

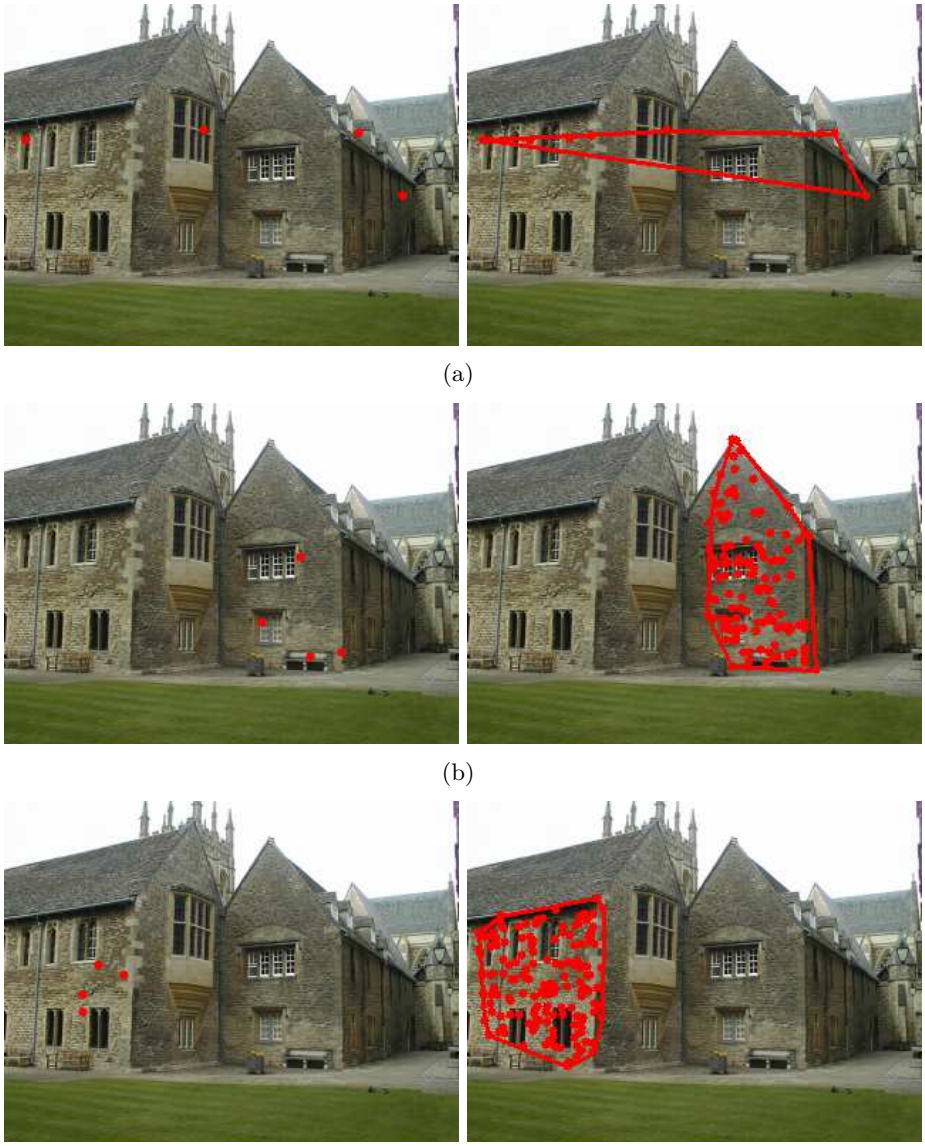
Nous souhaitons utiliser l'information de proximité spatiale. Les points d'une même structure étant souvent regroupés dans une même région de l'image, les échantillons formés à partir de données proches sont susceptibles de mener à une hypothèse pertinente. La première donnée est selectionnée aléatoirement. Les données suivantes sont sélectionnées avec une probabilité dépendant de la distance du point à l'échantillon en cours de construction. On définit la distance d'un point p à un ensemble de points $D = \{d_i\}$ de la manière suivante :

$$d_{spatiale}(p, D) = \max_i(\|p - d_i\|) \tag{4.4}$$

Etant donné l'échantillon $s = \{s_i\}_{i=1}^{taille<m}$ en cours de construction, la probabilité conditionnelle $P(p_j|s)$ de sélectionner p_j pour compléter s suit la relation suivante :

$$d_{spatiale}(p_1, s) \leq d_{spatiale}(p_2, s) \Rightarrow P(p_1|s) \geq P(p_2|s) \tag{4.5}$$

p_1 et p_2 étant deux points du jeu de données. L'algorithme 3 rappelle les étapes de BetaSAC, avec l'utilisation d'informations de proximité spatiale. La figure 4.8 montre les résultats obtenus avec une procédure aléatoire et en utilisant des critères de cohérence spatiale. Ces exemples illustrent les avantages d'une telle procédure. Les performances de la méthode sont démontrées dans la partie 4.6.



(a)

(b)

(c)

FIGURE 4.8 – **(a)** Echantillonnage aléatoire. **(b)** et **(c)** Echantillonnage utilisant la cohérence spatiale. La colonne de gauche affiche les 4 points de l'échantillon. La colonne de droite affiche les *inliers* du modèle calculé à partir de l'échantillon.

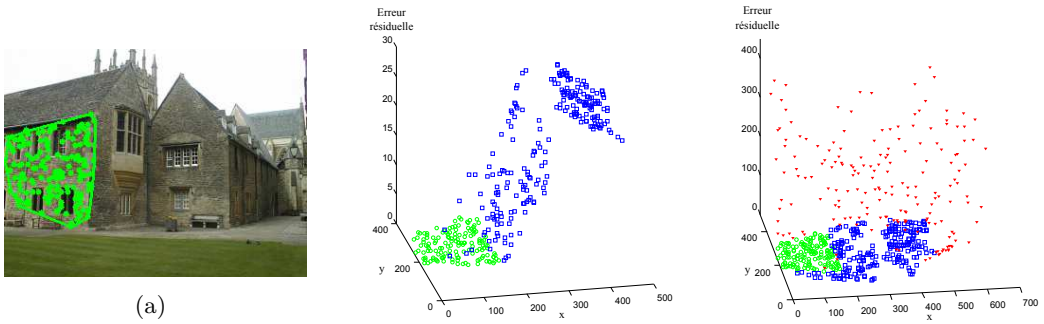
4.3.5 BetaSAC guidé : tri basé sur l'information tirée des itérations précédentes

On propose ici d'utiliser les informations que peuvent apporter les tirages précédents pour guider la sélection des échantillons suivants. Chaque hypothèse contient une information importante : l'erreur résiduelle des points par rapport au modèle généré. Pour chaque modèle généré, on peut classer les points en deux groupes selon la valeur de leur résidu. Si le résidu r est inférieur à un seuil τ, le point est *inlier*, sinon c'est un *outlier*. Dans le cas de multiples structures, ces *outliers* ont deux interprétations possibles. Elles peuvent être des données erronées provenant d'une

Algorithm 3 Tirage d'un échantillon s de taille minimale en utilisant une information de proximité spatiale

Require: t compteur d'itération, m taille minimale d'un échantillon

1: $s \leftarrow \emptyset$
2: **for** $l = 1$ to m **do**
3: Sélectionner n données selon une distribution uniforme.
4: Trier les n données par rapport à l'échantillon, avec la distance (équation 4.4).
5: Choisir a, la $1^{ière}$ donnée des n données triées.
6: $s \leftarrow s \cup a$
7: **end for**



(a) (b) (c)

FIGURE 4.9 – (a) *Inliers* du modèle généré. (b) Erreur résiduelle des correspondances en fonction des coordonnées des points, en vert les *inliers*, en bleu les *pseudo-outliers*. (b) Erreur résiduelle de l'ensemble des correspondances, avec les correspondances aberrantes (en rouge).

erreur de mesure ou bien des données consistantes avec une autre paramétrisation. Ces points sont appelés *pseudo-outliers*.

Comme l'illustre la figure 4.9, l'ensemble des *pseudo-outliers* correspondant à une paramétrisation distincte semblent avoir des résidus proches : alors que les *outliers* ont une erreur qui peut être très élevée, les *pseudo-outliers* sont regroupés dans l'espace formé par les coordonnées x et y du point dans la $1^{ière}$ image et l'erreur résiduelle au modèle. Sur la figure 4.9 (b), on distingue même une forme 3D proche de la forme que le bâtiment pourrait avoir. Nous montrons dans la section suivante comment utiliser l'erreur résiduelle comme un *a priori* pour générer des échantillons consistants.

4.3.5.1 Distance $d_{Résidus}$ exploitant l'information des itérations précédentes

Nous partons du principe que, quel que soit le modèle généré, deux points de résidus proches ont plus de chance d'appartenir à la même structure. Nous définissons la distance entre deux points par une distance entre résidus. Pour éviter de privilégier un modèle, nous utilisons l'information provenant de toutes les hypothèses déjà

générées.

On suppose que N_t hypothèses ont déjà été générées et on veut guider les prochains tirages. On forme pour chaque point p_i l'histogramme des résidus mesurés par rapport à chaque hypothèse :

$$\mathbf{r}^{(i)} = \left[r_1^{(i)} \; r_2^{(i)} \; ... \; r_{N_t}^{(i)} \right] \tag{4.6}$$

On définit le vecteur d'appartenance On définit alors la distance entre deux points p_1 et p_2 comme la distance entre leurs histogrammes respectifs de la manière suivante :

$$d_{Résidus}(p_1, p_2) = \sum_{j=1}^{M} \left\| r_j^{(1)} - r_j^{(2)} \right\| \tag{4.7}$$

On définit de la même façon la distance d'un point p à un ensemble de points $D = \{d_i\}$:

$$d_{Résidus}(p, D) = \max_i (d_{Résidus}(p, d_i)) \tag{4.8}$$

De la même manière que dans la partie 4.3.4, la distance est utilisée dans une stratégie BetaSAC (voir algorithme 3) pour former des échantillons de taille minimale. Les performances de la méthode sont évaluées dans la partie 4.6.

4.4 Optimisation locale

L'étape d'optimisation locale permet d'obtenir une paramétrisation plus précise du modèle recherché. La probabilité de former un échantillon sans données aberrantes peut s'approximer par ϵ^n, où ϵ est la proportion de points correspondant effectivement au modèle et n la taille de l'échantillon. Nous formons donc des échantillons de taille minimale pour maximiser cette probabilité. Cependant, de tels échantillons conduisent à des modèles imprécis qui ne coïncident pas forcément avec l'ensemble des *inliers* d'une structure. Nous utilisons la méthode d'optimisation décrite dans [Chum *et al.*, 2003]. A chaque nouvelle hypothèse, les paramètres du modèle sont réestimés en utilisant toutes les correspondances dont la distance au modèle est inférieure à $K * \theta$. K est un entier. θ est le seuil séparant les *inliers* des *outliers*. Cette étape est répétée en réduisant le seuil progressivement (on réduit K de 1) jusqu'à ce qu'il soit égal à θ.

En pratique, toutes les hypothèses ne sont pas nécessairement optimisées car la procédure prendrait alors K fois plus de temps. Nous évaluons les hypothèses générées suivant le critère suivant : une hypothèse est jugée prometteuse et donc candidate à l'optimisation si son support géométrique est en partie disjoint des supports géométriques des modèles déjà trouvés. Une hypothèse est optimisée si plus de 50% du support géométrique correspondant est disjoint des supports des structures déjà trouvées.

4.5 Fusion des hypothèses

4.5.1 Rejet des structures hypothétiques inconsistantes

Nous présentons dans cette partie comment sont évaluées les structures hypothétiques. Le critère de base de RANSAC pour comparer les structures hypothétiques est le nombre de points appartenant à cette structure. Ce critère n'est pas adapté à la détection de multiples structures. Ceci est dû au fait que certains points peuvent accidentellement se trouver sur une fausse structure. De plus, les petites structures reposant sur un faible nombre de points se trouvent désavantagées.

Nous évaluons chaque modèle hypothétique en estimant la variance du bruit des données en accord avec le modèle en question. Les données sont altérées par un bruit de mesure. Une correspondance en accord avec le modèle recherché (*inlier*) ne vérifie pas exactement la paramétrisation. On dit qu'elle est distante du modèle et cette distance est caractérisée par l'erreur résiduelle. L'analyse statistique de l'erreur résiduelle permet d'estimer la densité de probabilité de la distribution du bruit. En considérant que le bruit est un bruit gaussien de moyenne nulle, l'estimation de la variance de la distribution permet de définir un seuil d'erreur séparant les *inliers* des *outliers*. Plus ce seuil est faible, plus le modèle est fiable. Par conséquent, nous gardons une hypothèse si la variance du bruit estimée à partir des résidus relatifs à ce modèle est inférieure à un certain seuil. Notons que cette méthode analyse une distribution de scalaires. C'est donc la même méthode qui est utilisée, que l'on recherche une homographie ou une matrice fondamentale.

Dans l'application visée, les données contiennent très souvent plus de 50% de données inconsistantes (données erronées et données consistantes avec un autre modèle). Les estimateurs classiques ne sont pas applicables. Nous appliquons l'estimateur TSSE (*Two Step Scale Estimator*) non paramétrique décrit dans [Wang et Suter, 2004]. Les résidus de l'ensemble des points sont calculés avec la distance de Sampson (voir chapitre 2, section A.2) et classés par ordre croissant. L'analyse *mean-shift* de cette distribution estime la fonction de densité de probabilité. En considérant que l'erreur des *inliers* est de moyenne nulle, le premier minimum de la fonction (le plus proche de zéro) permet de séparer les *inliers* des *outliers*. La variance est alors réévaluée avec l'ensemble des *inliers* ainsi formé. La méthode est non paramétrique et estime la variance sans aucune connaissance *a priori*.

Nous vérifions les performances de l'estimateur de la variance sur les ensembles de la figure 4.10 : nous calculons les paramètres de chacune des homographies présentes avec l'ensemble des points appartenant à la structure. Nous connaissons en effet le support géométrique de chaque structure dans chacune des images. La variance du bruit est estimée à partir de la distribution de l'erreur résiduelle (distance de Sampson). Les données de départ sont précises et ne contiennent pas de données aberrantes. Nous répétons l'expérience en ajoutant aux données un bruit gaussien de variance comprise entre 0.05 et 2.5. Les résultats sont résumés figure 4.11 (a). L'estimateur obtient de bons résultats pour les ensembles 2 et 5 qui contiennent un plus grand nombre de points. Pour les autres structures, les performances diminuent rapidement avec l'augmentation du bruit. D'autre part, la variance calculée à partir

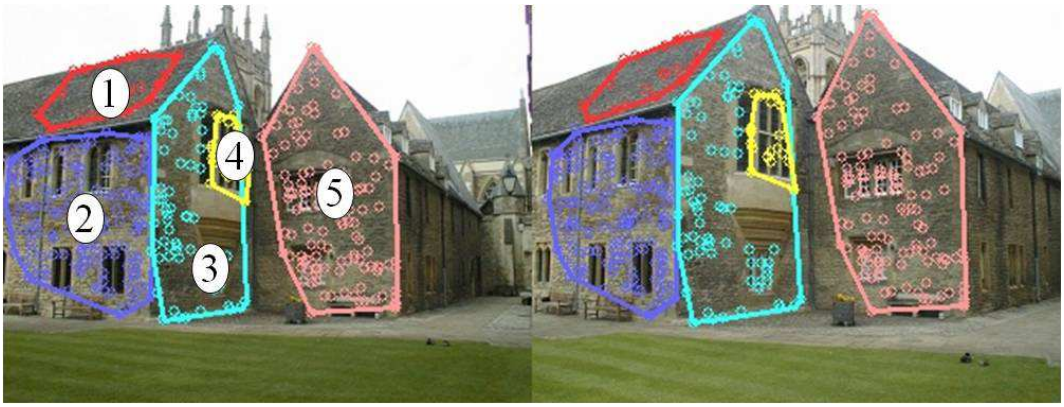


FIGURE 4.10 – Données utilisées pour évaluer l'estimateur TSSE. Les ensembles numérotés de 1 à 5 contiennent respectivement 22, 145, 69, 18 et 118 points.

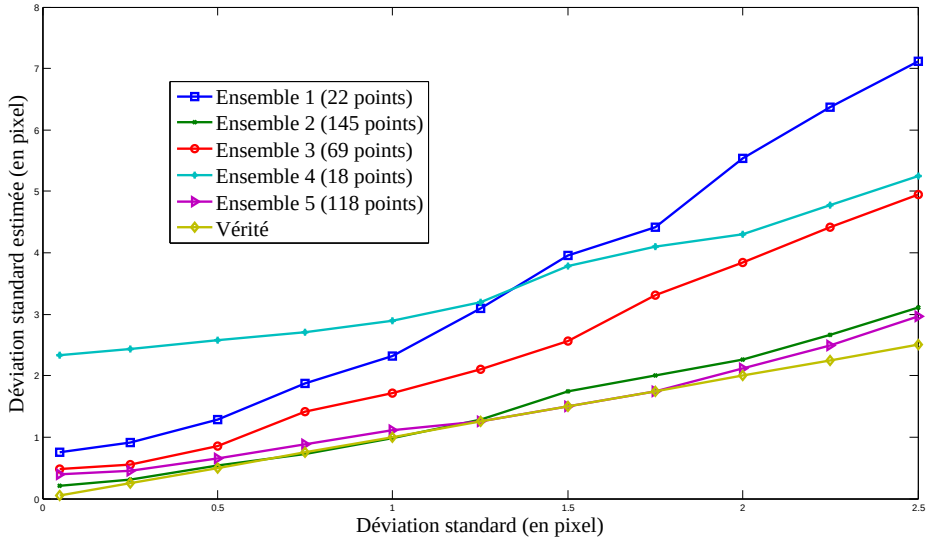
de l'ensemble 4 est toujours surestimée, quelque soit la valeur du bruit ajouté. Ceci est dû au fait que la structure correspondante est très proche de la structure reposant sur l'ensemble 3. Les deux modèles sont difficiles à séparer et l'estimation est moins précise.

Dans une deuxième expérience, nous évaluons la performance de l'estimateur relativement au nombre de données aberrantes présentes (figure 4.11 (b)). Un nombre croissant d'*outliers* est ajouté à l'ensemble des données (jusqu'à 450). L'estimateur échoue pour le calcul de la variance à partir des résidus de l'ensemble 1. En effet, pour les structures reposant sur un faible nombre de points, il y a plus de chance qu'un groupe d'*outliers* générés aléatoirement viennent perturber la segmentation de la structure. Les résultats concernant l'ensemble 4 sont moins perturbés car l'estimateur fusionne les *inliers* avec ceux de l'ensemble 3.

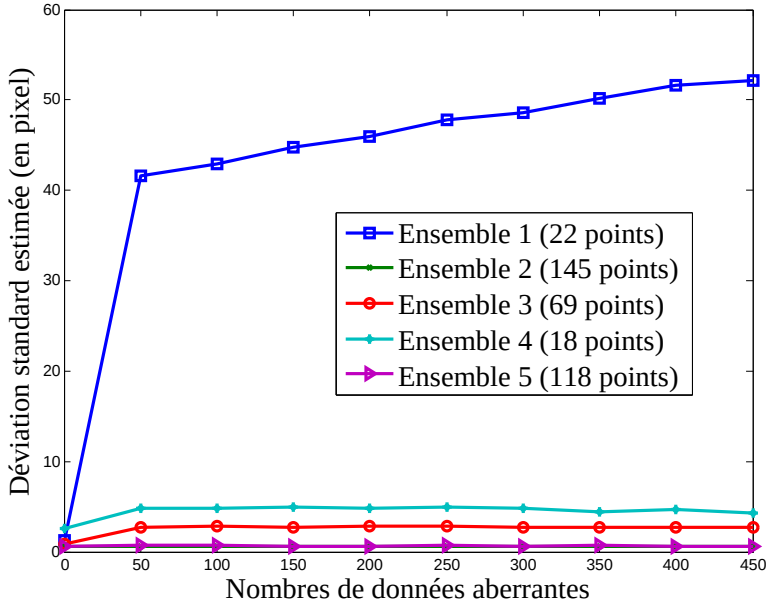
L'utilisation de l'estimateur TSSE présente quelques inconvénients. Il est difficile de séparer deux modèles proches et les modèles reposant sur des structures proportionnellement plus petites sont mal évalués en présence de données inconsistantes. Néanmoins, il obtient de meilleurs résultats que les estimateurs habituels. A titre de comparaison, les variances calculées avec l'estimateur utilisant la formule MAD (*Median Absolut Deviation*, formule 4.9) sont données dans le tableau 4.4. La variance réelle est de $0,2$, les valeurs données par MAD sont plus élevés. Les résultats montrent que l'estimateur échoue lorsque plusieurs structures sont présentes.

En conclusion, l'estimation de la variance du bruit permet d'évaluer rapidement la plausibilité d'une hypothèse. Cette étape améliore les performances de l'algorithme de détection de multiples structures et réduit considérablement le temps de calcul. Les hypothèses à partir desquelles on estime une variance supérieure à θ sont non pertinentes et sont automatiquement rejetées.

$$\sigma_{MAD} = 1.4826 * median_i(|X_i - median_j(X_j)|) \tag{4.9}$$



(a)



(b)

FIGURE 4.11 – Performances de l'estimateur TSSE avec la variation de la variance réelle (a) et du nombre de données erronées présentes (b).

Ensemble	1	2	3	4	5
$\sigma_{réelle}$	0,2	0,2	0,2	0,2	0,2
σ_{MAD}	19,1	16,9	13,1	11,04	16,21

TABLE 4.4 – Résultats de l'estimation de la variance du bruit avec l'estimateur MAD (bruit gaussien de variance $0,2$, aucune donnée abérrante).

4.5.2 Fusion des hypothèses avec les modèles trouvés

La collection de structures hypothétiques $\{S^t\}_{Hyp}$ est combiné avec $\{S^{t-1}\}$ pour garder les meilleurs modèles. Les ensembles de points de l'union $\{S^t\}_{Hyp} \bigcup \{S^{t-1}\}$ sont triés par taille décroissante. Un ensemble est ajouté à $\{S^t\}$ s'il est disjoint des ensembles déjà présents dans $\{S^t\}$. Cette procédure est répétée jusqu'à ne plus trouver d'ensembles disjoints. Deux ensembles sont considérés disjoints si le taux d'*inliers* en commun ne dépasse pas un seuil défini *a priori*. Dans nos expériences nous fixons ce seuil à 10%.

4.6 Evaluation des méthodes d'échantillonage proposées

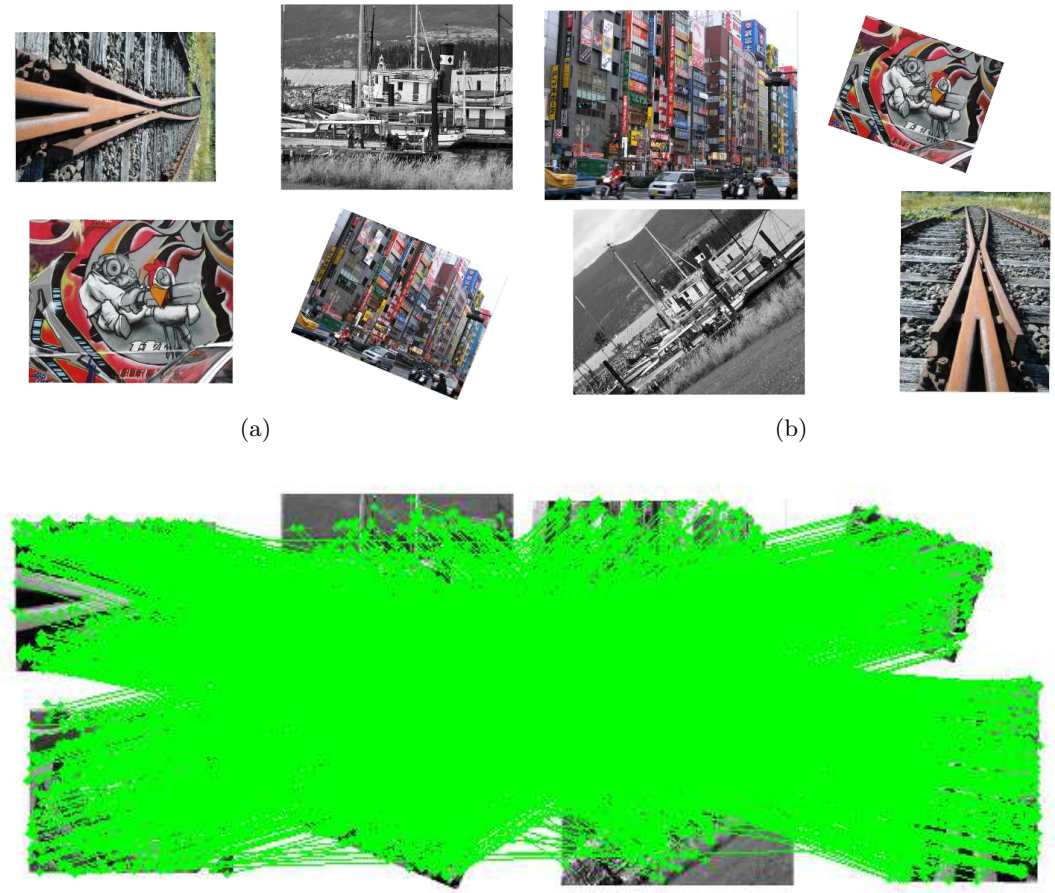
Nous réalisons une série d'expériences sur des données synthétiques et réelles pour évaluer les performances des méthodes d'échantillonage proposées et les comparer aux méthodes de l'état de l'art. Dans cette section, nous évaluons uniquement l'échantillonage et pas l'algorithme complet de détection de multiples structures. Une première expérience concerne l'estimation de similitudes. Les deux algorithmes proposés sont comparés à une méthode d'échantillonnage aléatoire classique (notée Random) et la méthode *Exp* (section 4.3.1). Les deux expériences suivantes permettent d'évaluer les performances des algorithmes sur l'estimation d'homographie et de matrice fondamentale. Les algorithmes sont comparés à l'échantillonnage aléatoire et à la méthode *Exp*. Le paramètre d'échelle de la méthode *Exp* (σ dans l'équation 4.3) prend la valeur de la distance moyenne au plus proche voisin. Toutes les expériences sont moyennées sur 50 exécutions.

4.6.1 Estimation de similitudes

Nous exposons ici l'étude des performances de notre algorithme sur le problème de l'estimation de multiples similitudes. C'est un cas simple et peu fréquent en pratique mais il permet de visualiser facilement les performances des algorithmes.

4.6.1.1 Problème et modélisation

Une similitude est la composition d'une isométrie avec une homothétie. Dans le cas d'une transformation euclidienne, une similitude a la représentation matricielle



(a) (b)

(c)

FIGURE 4.12 – Estimation de multiples similitudes. (c) Données d'entrée de l'algorithme : Correspondances entre les images (a) et (b).

suivante :

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.10}$$

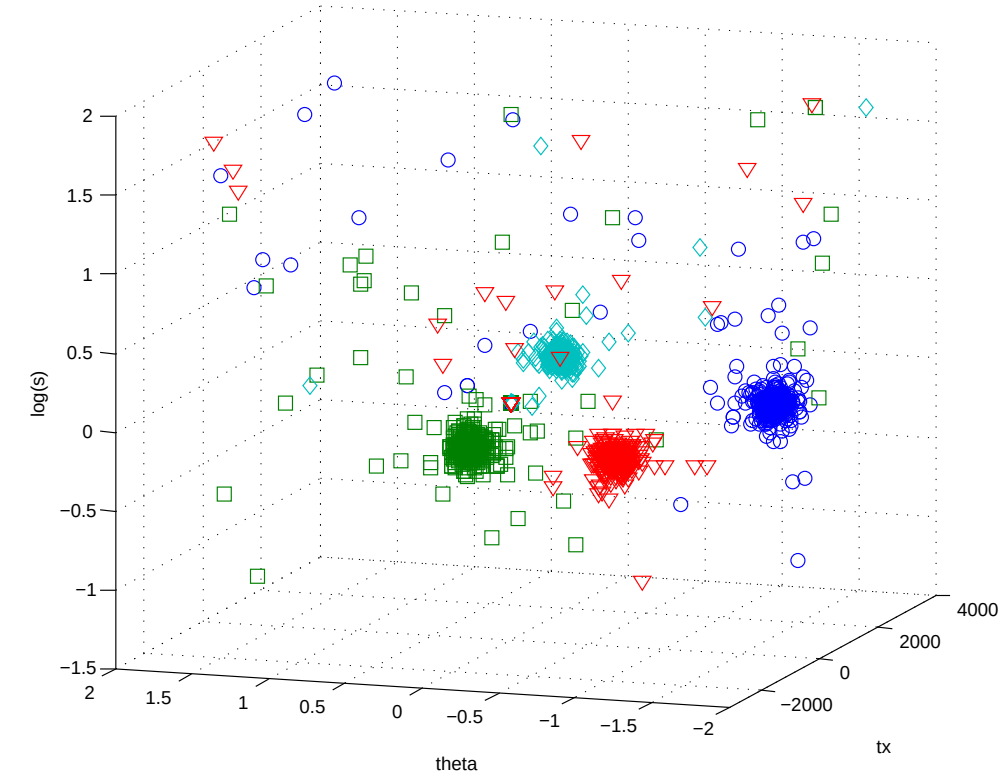
qu'on peut aussi écrire de manière plus concise :

$$\mathbf{p}' = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{p} = S\mathbf{p} \tag{4.11}$$

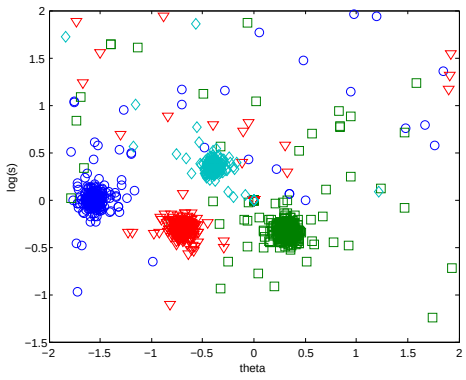
avec s, le rapport de l'homothétie, $\mathbf{R}$ la matrice de rotation d'angle θ et $\mathbf{t}$ le vecteur de translation. La matrice de similitude a quatre degrés de liberté, elle est déterminée par un ensemble minimal de deux correspondances.

4.6.1.2 Données

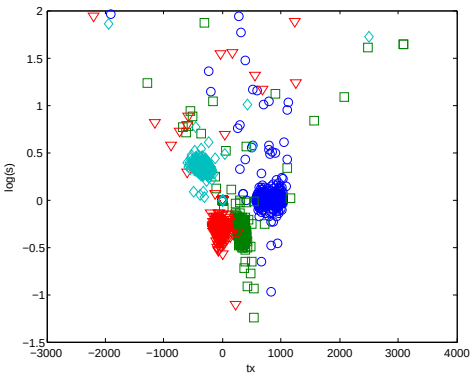
Nous évaluons les méthodes d'échantillonnage sur la paire d'images de la figure 4.12. Il s'agit en fait de quatre images déformées par quatre similitudes différentes.



(a)



(b)

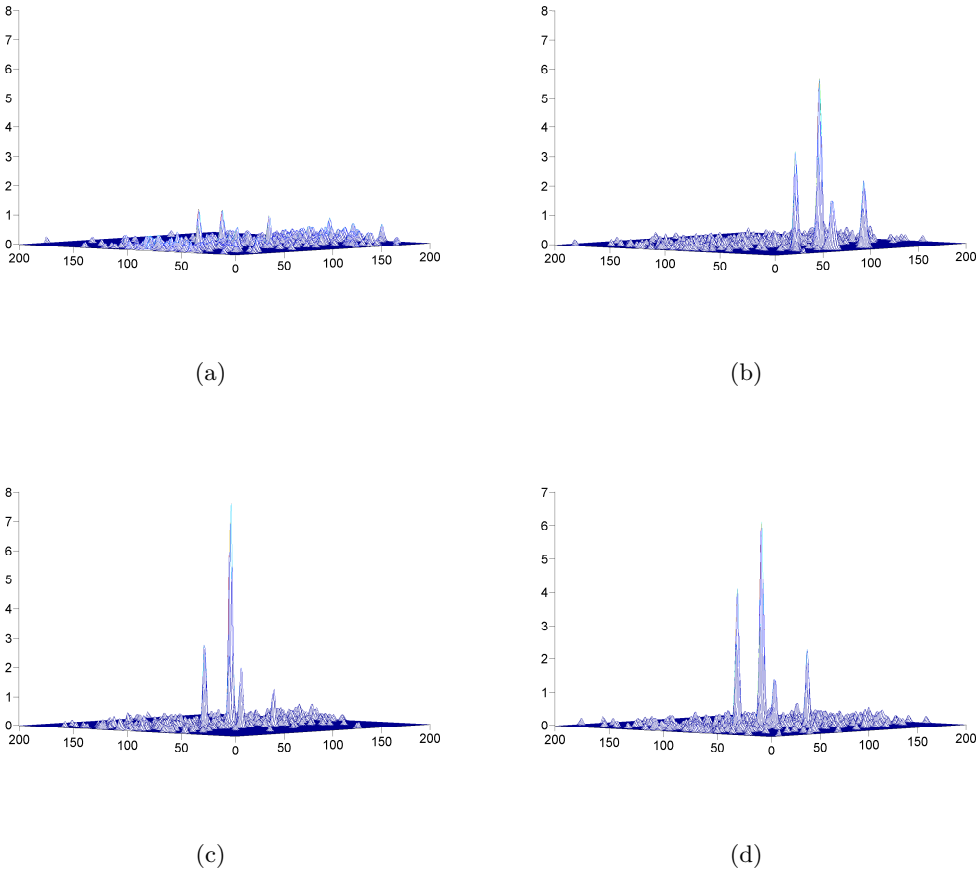


(c)

FIGURE 4.13 – Projection des hypothèses générées à partir des données de la figure 4.12 : quatre régions denses sont révélées dans l'espace des paramètres.

Les points SIFT sont détectés dans les deux images et mis en correspondances (figure 4.12 (c)). Pour définir une similitude plane, il suffit de fixer 4 paramètres correspondant à l'homothétie (s), la rotation (θ) et la translation (t_x et t_y). Cette expérience permet de visualiser les résultats dans l'espace des paramètres. La figure 4.13 montre les projections d'un ensemble de modèles dans les espaces paramètriques $(t_x, \theta, log(s))$, $(\theta, log(s))$, $(t_x, log(s))$. Pour chaque transformation, les paramétrisations sont calculées à partir des correspondances de points entre les deux images

correspondantes. Chaque point représente une transformation. On observe quatre régions denses correspondant aux quatre transformations présentes. En effet, lorsque la transformation entre deux groupes de points est effectivement une similitude, S a la même valeur pour tous les couples de bonnes correspondances, au bruit de mesure près. Les paramétrisations obtenues à partir d'ensembles consistants forment un amas dans l'espace des paramètres.



(a) (b)

(c) (d)

FIGURE 4.14 – **Densité des paramètres** $(\theta, log(s))$ **pour chaque méthode.** (a) Echantillonnage aléatoire, (b) *Exp*, (c) BetaSAC spatial, (d) BetaSAC guidé.

4.6.1.3 Expérimentation

Nous comparons notre méthode à une méthode d'échantillonage aléatoire classique et à la méthode *Exp*. Pour chaque méthode, 500 hypothèses de similitudes sont générées et nous calculons la densité des paramètres $(\theta, log(s))$ à partir des modèles générés par chaque méthode. Les résultats sont résumés figure 4.14. Les hypothèses générées par nos algorithmes sont pertinentes : on observe quatre pics élevés correspondant aux quatre modèles existants. L'échantillonnage aléatoire génère beaucoup d'hypothèses incorrectes et la méthode *Exp* n'atteint pas nos performances.

4.6.2 Estimation d'homographies

Nous exposons dans cette section une seconde série d'expériences sur l'estimation d'homographie. Les algorithmes sont évalués sur des données synthétiques et réelles.

4.6.2.1 Expériences sur des données synthétiques

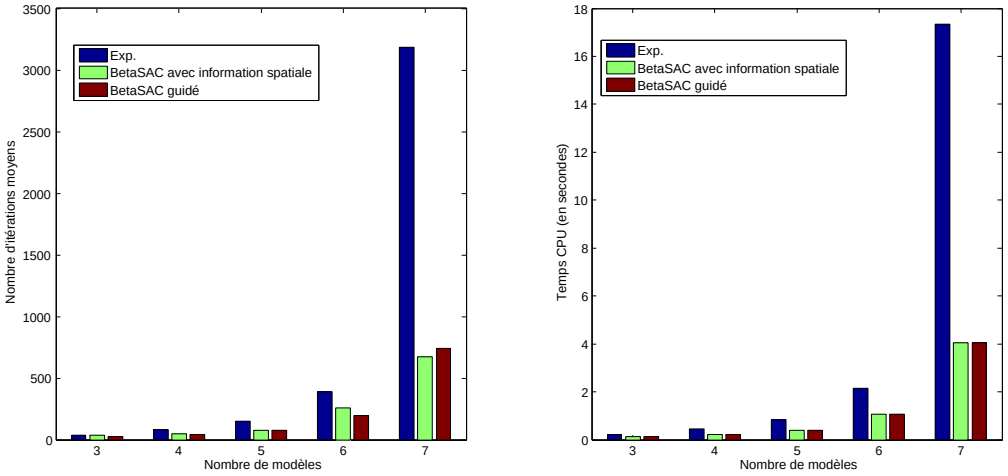
Dans une première expérience, les performances des algorithmes sont mesurées en faisant varier le nombre de modèles présents dans les données. 500 correspondances de points sont générées. Le nombre de modèles présents dans les données varie de 3 à 7. On ajoute un bruit gaussien d'écart-type $\sigma_{noise} = 0.5$ et un certain nombre de correspondances aberrantes (20%). Le nombre de points par structure varie pour évaluer notre algorithme sur des structures de tailles différentes. Le tableau 4.5 donne, pour chaque jeu de données, le nombre de points correspondant à chaque modèle. Les données sont labélisées : à chaque point est associé un numéro désignant

Données \ Modèle	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Données 1	221	108	171	0	0	0	0
Données 2	93	124	176	107	0	0	0
Données 3	91	146	98	68	97	0	0
Données 4	73	77	61	159	88	42	0
Données 5	104	36	44	39	109	89	79

TABLE 4.5 – Données synthétiques : nombre de points par structure.

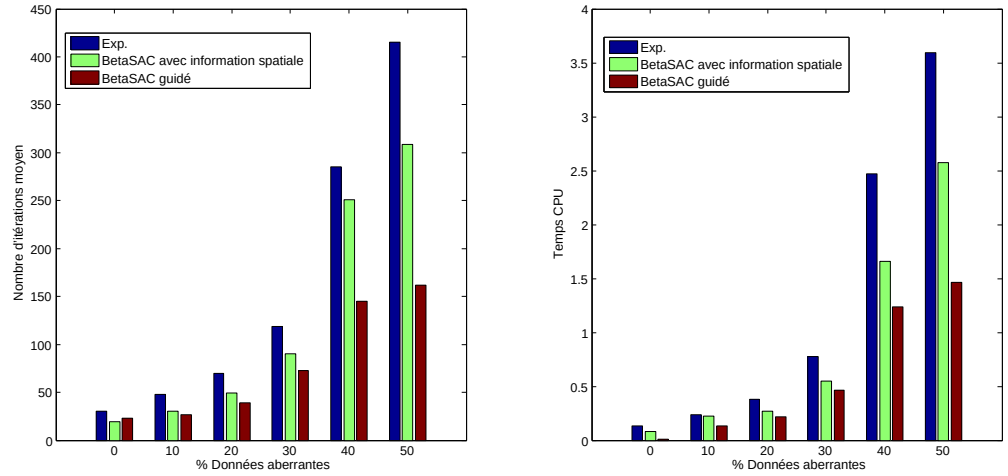
la structure à laquelle il appartient. Nous mesurons la performance en comptant le nombre d'itérations minimales nécessaires (et le temps CPU correspondant) pour générer au moins un échantillon consistant par modèle (*i.e.* constitué uniquement de points appartenant à la même structure). Les résultats sont résumés figure 4.15. Le graphe de gauche montre le nombre d'itérations moyen en faisant varier le nombre de modèles présents. Le graphe de droite donne le temps CPU moyen en secondes. Nous répétons cette expérience en fixant le nombre de modèles présents à 4 et en faisant varier le nombre de données abérrantes de 0% à 50%. Les résultats sont résumés figure 4.16. Les résultats de la méthode d'échantillonnage aléatoire ne sont pas affichés car les valeurs sont nettement plus élevées que pour les autres méthodes.

On peut noter que les stratégies BetaSAC sont plus efficaces que la méthode *Exp* couramment utilisée dans la littérature, notamment lorsque le nombre de modèles présents ou lorque le nombre de données aberrantes augmente. Les méthodes d'échantillonnage proposées génèrent des échantillons consistants plus rapidement, elles vont permettre d'accélérer des algorithmes de détection de multiples structures. La performance des stratégies BetaSAC est confirmée sur des données réelles dans la section suivante.



(a) (b)

FIGURE 4.15 – **Performances des différentes méthodes d'échantillonnage en fonction du nombre de modèles présents dans les données.** (a) et (b) : Nombre d'itérations et temps nécessaires pour générer au moins un échantillon consistant (i.e. constitué uniquement d'*inliers*) par modèle.



(a) (b)

FIGURE 4.16 – **Performances des différentes méthodes d'échantillonnage en présence de données aberrantes.** (a) et (b) : Nombre d'itérations et temps nécessaires pour générer au moins un échantillon consistant (i.e. constitué uniquement d'*inliers*) par modèle.

(a) (b)

(c) (d)

FIGURE 4.17 – Paires d'images utilisées pour le problème d'estimation de multiples homographies. Les différentes couleurs correspondent aux supports des differents plans.

Données		Random	Exp	BetaSAC Spatial	BetaSAC Guidé
College-1	I_1(163,34%)	13, 6	97, 1	143, 6	**216**
	I_2(168,35%)	14, 5	110, 2	152, 8	**260, 1**
	I_{Tot}	28, 1	207, 3	296, 4	**476, 1**
College-2	I_1(214,32%)	10, 5	80, 9	117, 4	**132**
	I_2(119,18%)	1, 2	18, 9	31, 5	**32, 7**
	I_3(75,11%)	0, 1	6, 9	**9, 2**	7, 1
	I_4(38,6%)	0	0, 3	**1, 4**	1
	I_{Tot}	11, 8	107	159, 5	**172, 8**
College-3	I_1(22,4%)	0	0	0.5	**0.6**
	I_2(145,26%)	4, 7	58	116	**148**
	I_3(69,12%)	0, 2	1, 4	7	**10**
	I_4(18,3%)	0	0	0.5	**0.5**
	I_5(118,21%)	2, 1	28	66	**103**
	I_{Tot}	7	87, 4	190	**262**
Johnson	I_1(20,3%)	0	0, 1	0, 2	**0.3**
	I_2(77,12%)	0, 2	4	9, 2	**9, 7**
	I_3(291,45%)	40, 9	144, 8	263, 3	**274, 8**
	I_4(69,11%)	0, 1	1, 6	4, 5	**5, 7**
	I_5(41,6%)	0	0, 4	2, 2	**2, 1**
	I_6(15,2%)	0	0	0	**0, 1**
	I_7(58,9%)	0, 1	7, 5	14, 8	**16**
	I_{Tot}	41, 3	158, 4	294, 2	**308, 7**

TABLE 4.6 – Performance des méthodes d'échantillonnage. A chaque exécution, 1000 échantillons sont générés. Le **nombre d'échantillons consistants formés** est donné pour chaque structure $I_i, i = 1, 2....$ La deuxième colonne donne le nombre de points par structure et le pourcentage correspondant.

4.6.2.2 Expériences sur des données réelles

Nous comparons les différentes méthodes sur quatre paires d'images stéréoscopiques réelles disponibles sur Internet [1], [2]. Les scènes photographiées contiennent plusieurs plans (de 2 à 7). La figure 4.17 montre les paires d'images utilisées et les structures présentes. Les différentes couleurs correspondent aux supports géométriques des differents plans dans les deux images similaires. Nous affichons sur la figure les données consistantes uniquement. Nous utilisons les correspondances de points fournies avec les données. Les données de l'université d'Oxford ne contiennent que les correspondances correctes, nous ajoutons donc artificiellement un certain nombre de correspondances incorrectes : elles représentent environ 40% des données. Ce taux de données incorrectes est élevé. En pratique, sur nos données réelles, il est proche de 20%. Les données sont labélisées : nous connaissons le support géométrique de chaque structure.

La performance des algorithmes est mesurée par le nombre d'échantillons consistants générés. Au cours d'une expérience, 1000 échantillons de quatre correspondances sont formés et les paramètres des homographies correspondantes sont estimés. On compte alors le nombre d'échantillons consistants. Cette expérience est répétée pour chaque méthode. Les résultats sont résumés dans le tableau 4.6. Les performances de BetaSAC vis-à-vis des autres méthodes sont confirmées sur données réelles. Néanmoins, on note aussi que toutes les méthodes échouent lorsque la structure repose sur un faible pourcentage des points (inférieur à 10%). Ces résultats suggèrent donc d'utiliser une méthode s'apparentant au RANSAC séquentiel : les points des modèles déjà trouvés sont retirés du jeu de données pour révéler les structures plus petites. Cette idée est reprise dans notre algorithme lors de la formation des hypothèses.

4.6.3 Estimation de matrices fondamentales

Nous exposons dans cette section l'application de nos algorithmes à l'estimation de matrices fondamentales.

Au cours d'une expérience, 1000 échantillons de sept correspondances sont formés. Nous comptons le nombre d'échantillons consistants. Les données sont des paires d'images disponibles sur internet [3]. Ce sont des observations d'une même scène prises à des instants différents (figure 4.18). Plusieurs objets (de 2 à 4) ont bougé dans la scène. Les paires d'images utilisées sont représentées sur la figure 4.18. Les résultats sont résumés figure 4.7. Les résultats montrent l'efficacité des stratégies BetaSAC, alors que l'échantillonage aléatoire et la méthode *Exp* échouent à générer des échantillons consistants. Le nombre d'échantillons consistants générés est plus élévé pour ces deux méthodes. Nous notons aussi que les informations tirées des itérations précédentes sont plus discriminantes que l'information spatiale seule. Néanmoins, le tirage est plus lent car il nécessite de calculer la distance entre résidus

1. http ://www.robots.ox.ac.uk/ vgg/data/data-mview.html
2. http ://cs.adelaide.edu.au/ hwong/doku.php ?id=data
3. http ://cs.adelaide.edu.au/ hwong/doku.php ?id=data

(a) (b)

(c)

FIGURE 4.18 – Paires d'images utilisées pour le problème d'estimation de multiples gémoétries épipolaires. Les différentes couleurs correspondent aux points des differents mouvements. Nous affichons ici les données consistantes uniquement.

Données		Random	Exp	BetaSAC Spatial	BetaSAC Guidé
2 objets	I-1(78,31%)	$0,3$	$5,6$	$110,9$	$\mathbf{214,8}$
	I-2(72,29%)	$0,2$	$4,4$	$142,9$	$\mathbf{196,8}$
	I-Tot	$0,5$	10	$253,8$	$\mathbf{411,6}$
3 objets	I-1(19,12%)	0	0	$3,8$	$5,6$
	I-2(33,20%)	0	$0,1$	$69,4$	$73,6$
	I-3(53,32%)	$0,3$	$10,2$	$208,8$	$\mathbf{248,4}$
	I-Tot	$0,3$	$10,3$	282	$\mathbf{327,6}$
4 objets	I-1(71,22%)	0	$4,9$	$67,8$	$\mathbf{94,5}$
	I-2(49,15%)	0	$1,9$	$17,7$	$\mathbf{32,8}$
	I-3(38,12%)	0	$0,1$	$6,3$	**11**
	I-4(81,25%)	0	$6,3$	$48,7$	$\mathbf{144,9}$
	I-Tot	0	$13,2$	140.5	$\mathbf{283,2}$

TABLE 4.7 – Performance des méthodes d'échantillonnage. A chaque exécution, 1000 échantillons sont générés. Le **nombre d'échantillons consistants formés** est donné pour chaque structure $I_i, i = 1, 2....$ La deuxième colonne donne le nombre de points par structure et le pourcentage correspondant.

définie dans la partie 4.3.5.

4.7 Evaluation de l'algorithme de détection de multiples structures

Nous présentons dans cette section les résultats de l'algorithme complet de détection de multiples structures que nous comparons aux méthodes de partitionnement

J-Linkage [Toldo et Fusiello, 2008] et de minimisation d'énergie [Isack et Boykov, 2012].

4.7.1 Données synthétiques

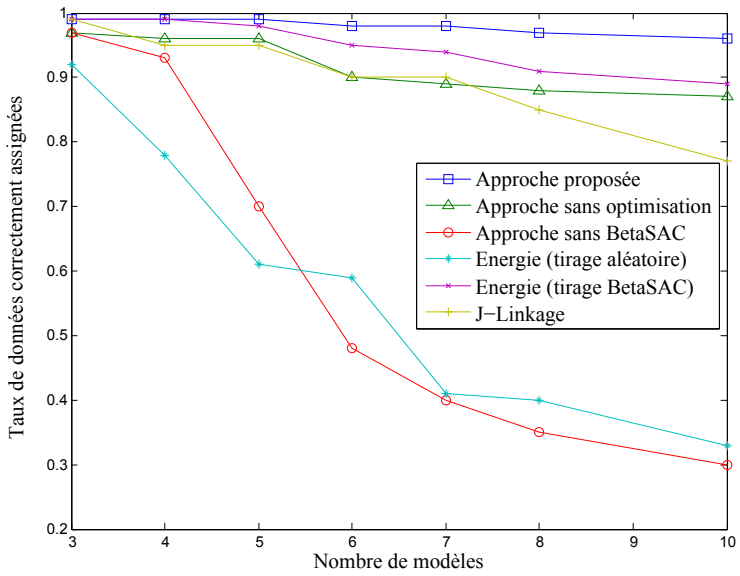


FIGURE 4.19 – Performances des différentes méthodes en faisant varier le nombre de modèles présents.

Nous évaluons tout d'abord les différentes méthodes sur des données synthétiques sur le problème de l'estimation de multiples homographies. Ce sont les mêmes données que celles de la partie 4.6.2.

500 correspondances de points sont générées.

Dans la première expérience, nous faisons varier le nombre d'homographies de 3 à 7. Le bruit gaussien a une variance $\sigma_{noise} = 0.5$ et un certain nombre de correspondances aberrantes sont ajoutées (20%).

Dans la seconde expérience, le nombre de modèles présent est fixé à 4 et nous faisons varier la proportion de données abérrantes de 0% à 50%. Les résultats sont moyennés sur 100 exécutions. Nous mesurons les performances des différentes méthodes par le pourcentage de données correctement associées. Pour chaque méthode, le nombre d'hypothèses générées est fixé à 500. Les résultats sont résumés dans les diagrammes 4.19 et 4.20.

Nous avons évalué les différentes étapes de notre algorithme en répétant l'expérience avec l'algorithme complet, l'algorithme sans l'étape d'optimisation et l'algorithme sans échantillonnage adapté. On peut noter que la stratégie d'échantillonnage

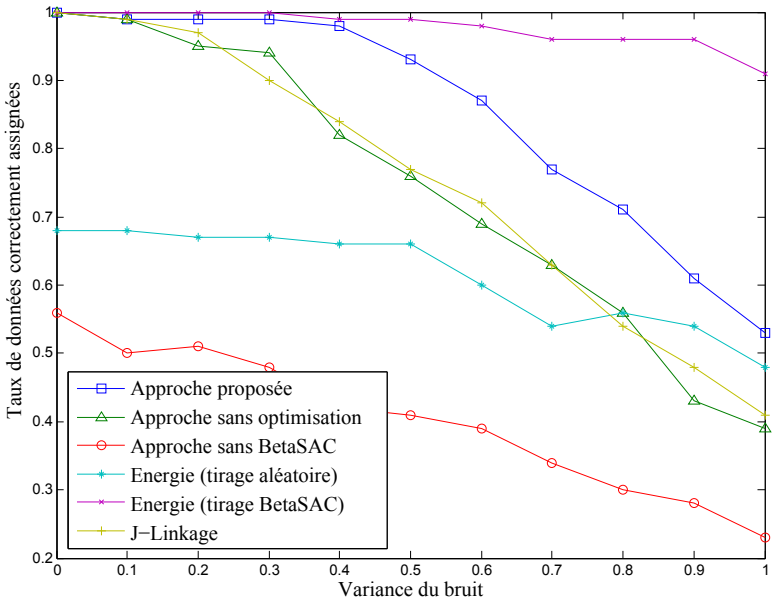


FIGURE 4.20 – Performances des différentes méthodes en faisant varier le bruit de mesure.

BetaSAC améliore considérablement les performances. En présence de bruit, l'étape d'optimisation est indispensable.

Nous avons aussi évalué sur les même données les méthodes de [Toldo et Fusiello, 2008] (J-Linkage) et [Isack et Boykov, 2012] (Minimisation d'énergie). Nous évaluons cette dernière méthode avec un échantillonnage aléatoire classique et notre stratégie d'échantillonnage utilisant la cohérence spatiale. Notre algorithme obtient de meilleurs résultats que ces deux méthodes. Cependant, en présence de bruit, la combinaison de la stratégie BetaSAC avec la régularisation de l'energie de [Isack et Boykov, 2012] est plus performante.

4.7.2 Données réelles bruitées

Méthodes \ Données	Oxf.-1	Oxf.-1	Oxf.-1	Johnson
Approche proposée	88%	72%	**79%**	**65%**
J-Linkage	80%	64%	61%	56%
Energie-Exp	83%	49%	50%	50%
Energie-BetaSAC	**92%**	**76%**	71%	63%

TABLE 4.8 – **Estimation de multiples homographies.** Pourcentage de données correctement associées

Dans cette partie nous présentons les performances des différents algorithmes évalués et comparés sur des données réelles sur les problèmes d'estimation d'ho-

Méthodes \ Données	2 objets	3 objets	4 objets
Approche proposée	83%	80%	78%
Energie-Exp	73%	77%	64%

TABLE 4.9 – **Estimation de multiples matrices fondamentales.** Pourcentage de données correctement associées

mographie et de matrice fondamentale. Les méthodes sont évaluées sur les données présentées figure 4.17 et figure 4.18. Les données sont bruitées avec un bruit gaussien de variance $\sigma_{Bruit} = 0.5$ et environ 20% des données sont aberrantes. Les différentes structures et les ensembles de points correspondants sont connus. Nous mesurons la performance avec le pourcentage de points correctement associés. Les résultats sont résumés dans le tableau 4.8 pour l'estimation d'homographie et dans le tableau 4.9 pour l'estimation de matrice fondamentale. L'algorithme proposé obtient de bons résultats mais ses performances sont parfois dépassées par la méthode [Isack et Boykov, 2012] combinée avec la stratégie d'échantillonnage BetaSAC.

4.8 Conclusion

Nous avons proposé dans ce chapitre, un nouvel algorithme de détection de multiples modèles. Ce dernier est une étape essentielle à l'extraction automatique d'objets présentée dans le chapitre 3. La méthode proposée combine une stratégie d'échantillonnage adaptée s'appuyant sur l'algorithme BetaSAC et une étape d'optimisation permettant d'améliorer la qualité des modèles trouvés. D'autre part, elle rejette automatiquement les hypothèses non pertinentes, celles pour lesquelles la variance estimée du bruit est supérieure à un certain seuil. L'intérêt est que les modèles trouvés sont toujours pertinents quel que soit le temps de calcul dédié à l'étape de détection de multiples modèles.

Nous avons appliqué l'algorithme à la détection de multiples homographies et à la détection de multiples matrices fondamentales. Les résultats obtenus sont en général meilleurs que ceux obtenus avec les approches de l'état de l'art. La détection de multiples homographies est fonctionnelle et utilisée avec succès dans le chapitre 3.

L'application de l'algorithme à la détection de multiples matrices fondamentales n'a pas obtenu de résultats probants. Nous avons constaté dans les expériences que ce problème est plus difficile. Ceci est en partie dû au fait qu'il n'existe pas de bijection reliant deux points mis en correspondance. La matrice fondamentale transforme un point en une droite. Le calcul d'erreur est moins précis que pour une homographie et les modèles obtenus manquent de précision. Ces résultats pourraient être améliorées en combinant les données extraites de plusieurs images successives.

Chapitre 5

Amélioration d'une méthode de SLAM topologique

Contenu du chapitre

Nous présentons dans ce chapitre comment utiliser la modélisation de l'environnement que nous avons proposé dans le chapitre 3 pour améliorer une méthode de SLAM topologique. Nous introduisons dans un premier temps le formalisme probabiliste sur lequel repose la méthode de SLAM topologique, puis les modifications apportées et les performances de notre approche évaluées sur des données réelles.

5.1 Formalisme bayésien pour la localisation par vision

Dans sa forme probabiliste, le problème du SLAM revient à calculer à chaque instant t la distribution de probabilité :

$$p\left(x_t|Z^t, x_0\right) \tag{5.1}$$

où x_t est la position de la caméra à l'instant t, $Z^t = \{Z_1, Z_2, ..., Z_t\}$ sont les observations acquises jusqu'au temps t et x_0 est la position initiale de la caméra. Les auteurs de [Cummins et Newman, 2007] et [Angeli *et al.*, 2008] proposent d'utiliser un filtrage bayésien. Si n lieux sont déjà enregistrés dans la carte topologique, la position de la caméra est estimée au sens du maximum *a posteriori* et correspond au lieu L_j dont l'index vérifie :

$$j = \underset{i=-1,1,...,n}{\operatorname{argmax}}\, p\left(L_i|Z^t\right) \tag{5.2}$$

Le résultat $j = -1$ signifie qu'aucun lieu n'a été reconnu et qu'il faut mettre à jour la carte en ajoutant un nouveau lieu. D'après la loi de Bayes, la probabilité d'être dans le lieu L_i s'écrit :

$$p\left(L_i|Z^t\right) = \frac{p\left(Z_t|L_i\right) p\left(L_i|Z^{t-1}\right)}{p\left(Z_t|Z^{t-1}\right)} \tag{5.3}$$

5.1.1 Prédiction

$p\left(L_i|Z^{(k-1)}\right)$ est le terme de prédiction : c'est la probabilité a priori d'être dans le lieu L_i connaissant les observations jusqu'au temp $t-1$. Une prédiction du lieu au temps k peut être obtenue à partir de la position antérieure et d'un modèle de mouvement simple. Dans le problème de la localisation par vision, il n'y a pas de données odométriques disponibles pour définir un modèle de mouvement. Dans les approches [Cummins et Newman, 2007] et [Angeli *et al.*, 2008], les auteurs font l'hypothèse que deux images successives dans la séquence vidéo proviennent de lieux adjacents. Si la caméra est dans le lieu L_i au temps $t-1$ il est probable qu'elle soit dans L_i, ou dans le voisinage de L_i au temps t. Ces considérations sont prises en compte dans le calcul de la prédiction.

5.1.2 Normalisation

La probabilité *a posteriori* est estimée en multipliant la vraisemblance et le terme de prédiction et en normalisant par :

$$p\left(Z_t|Z^{t-1}\right) = \sum_{i=-1}^{N_l} p\left(Z_t|L_i\right) p\left(L_i|Z^{t-1}\right) \tag{5.4}$$

N_l est le nombre de lieux déjà enregistrés dans la carte. On obtient ainsi la densité de probabilité sur l'ensemble des lieux.

5.1.3 Vraisemblance de l'observation

$p\left(Z_t|L_i\right)$ est la vraisemblance de l'observation. Elle mesure l'adéquation entre l'observation au temps t et le modèle de chaque lieu. Chaque lieu est modélisé par un sac de mots visuels. Le calcul de vraisemblance revient donc à calculer un score de similarité entre le sac de mots visuels de l'observation courante et celui de chaque lieu déjà visité et enregistré dans la base de données (chapitre 2, section 2.2.3).

Cette méthode est utilisée par les auteurs [Cummins et Newman, 2007] et [Angeli *et al.*, 2008] qui font l'hypothèse d'un environnement statique. Elle présente des limites dans une scène dynamique. Un lieu peut ne pas être reconnu car l'apparence de la scène est trop altérée par le mouvement des objets et un nouveau lieu est ajouté à tort à la carte topologique. Par ailleurs, lorsqu'un objet saillant est déplacé dans un lieu différent, la méthode va reconnaître l'objet présent dans ce lieu et la détection est faussée. Des liens entre deux lieux non adjacents sont alors ajoutés à tort à la carte topologique. La figure 5.1.3 présente des cas limites de ce type observés lors de nos expérimentations avec une méthode de reconnaissance de lieu reposant sur le critère du maximum de vraisemblance. Chaque ligne présente deux images similaires. On observe à gauche l'image courante et à droite l'image de la base de données. Les points bleus sont les points validés par le calcul de matrice fondamentale entre les deux vues. Les points rouges sont les *outliers*.

Nous souhaitons modifier le calcul de vraisemblance pour adapter la méthode à notre contexte d'environnement dynamique.

5.2 Nouvelle formulation pour le calcul de vraisemblance

Nous présentons ici une nouvelle formulation permettant de prendre en compte la connaissance des objets dynamique de la scène dans le calcul de vraisemblance.

TABLE 5.1 – Cas limites de la méthode de reconnaissance de lieu dans un environnement dynamique

5.2.1 Intégration de la notion d'objet dans le modèle

Dans un environnement statique, nous pouvons définir le modèle d'apparance du lieu par l'ensemble des mots observés dans ce lieu. Etant donné un vocabulaire de mots visuels V de taille $|V|$ préalablement appris, le modèle d'un lieu est illustré par la figure 5.1. L'observation capturée au temps k est notée $Z_k = z_1, ..., z_{|V|}$.

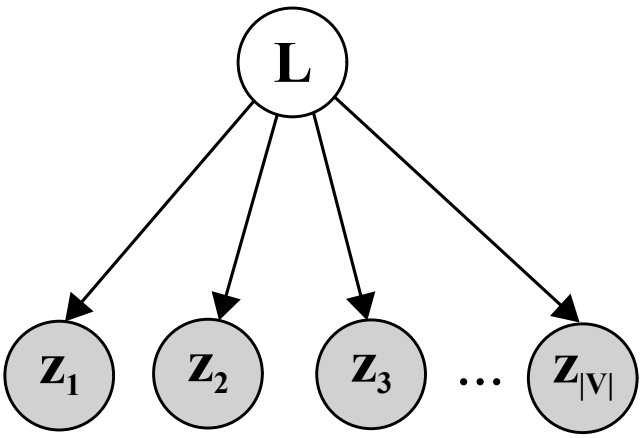


FIGURE 5.1 – Modèle d'un lieu dans un environnement statique : $L = \left\{z_1, ..., z_{|V|}\right\}$

La composante z_i est une variable binaire qui prend la valeur 1 si le $i^{ième}$ mot du vocabulaire est présent dans l'observation. La vraisemblance de l'observation se calcule alors de la manière suivante, en supposant l'indépendance entre les variables

d'observations :

$$p\left(Z_k|L_i\right) = \prod_{j=1}^{|V|} p\left(z_j|S, L_i\right). \tag{5.5}$$

Dans un environnement dynamique, nous définissons la scène comme une structure statique et un ensemble d'objets dynamiques. La connaissance des objets permet de mettre à jour le formalisme de reconnaissance de lieu. Les mots visuels de l'observation courante appartiennent soit à la structure statique du lieu, soit à l'un des objets. Le nouveau modèle graphique est illustré figure 5.2.

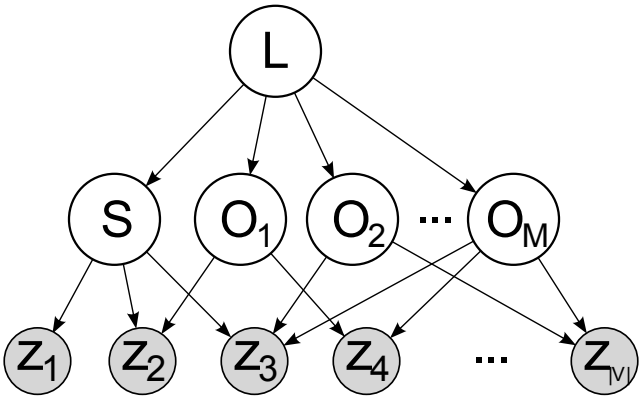


FIGURE 5.2 – Modèle d'un lieu dans un environnement dynamique : $S = \left\{z_{S_1}, ..., z_{S_{|V|}}\right\}$, pour tout objet i, $O_i = \left\{z_{O_{i1}}, ..., z_{O_{i|V|}}\right\}$ et $L = \{S, O_1, ..., O_M\}$.

On cherche à calculer la probabilité de se trouver dans le lieu L_i, ce qui revient à calculer la probabilité d'oberver la structure statique du lieu S_i et un ensemble d'objets $\{O_1, ..., O_M\}$:

$$p\left(L_i|Z\right) = p(S_i, O_1, ..., O_M|Z) \tag{5.6}$$

$$= p\left(L_i\right) p\left(S_i|L_i, Z\right) \prod_{l=1}^{M} p\left(O_l|L_i, Z\right) \tag{5.7}$$

$$= p\left(L_i\right) p\left(S_i|Z\right) \prod_{l=1}^{M} p\left(O_l|L_i, Z\right) \tag{5.8}$$

$\{O_l\}_{i=1}^{M}$ est l'ensemble des objets dynamiques connus. Les probabilités d'être dans la structure ou dans l'un des objets du lieu s'écrivent classiquement avec les probabilités d'occurence des mots visuels.

$$p\left(S_i|Z\right) = \prod_{j=1}^{|V|} p\left(z_j|S_i\right) \tag{5.9}$$

$$p\left(O_l|L_i, Z\right) = \prod_{j=1}^{|V|} p\left(z_j|O_l, L_i\right) \tag{5.10}$$

Dans la suite du chapitre, nous nous intéressons uniquement au calcul de vraisemblance de l'observation courante $p(Z|S_i, O_1, ..., O_M)$. Connaissant les lieux et les objets dynamiques de l'environnement, nous souhaitons déterminer la combinaison du lieu et des objets dynamiques présents qui correspondrait le plus à l'image courante. Nous verrons dans les expériences que ce procédé permet de rejeter les fausses reconnaissances de lieu qui occurrent si on fait l'hypothèse d'un milieu statique.

5.3 Modélisation des lieux et des objets dynamiques

Pour calculer la vraisemblance de l'observation, nous devons disposer du modèle d'apparence de chaque lieu et de chaque objet de l'environnement. Nous supposons ici que nous disposons de deux séquences vidéos prises dans un même environnement.

Nous avons vu dans le chapitre 3 comment détecter de manière automatique des objets dynamiques à partir de ces deux séquences. L'objet est alors représenté par un ensemble de points SURF. Nous apparions ces points dans les images précédentes et suivantes en vérifiant la cohérence géométrique. Cela permet d'obtenir les points de l'objet dans toutes les images de la séquence. Le modèle de l'objet est alors défini par l'ensemble des descripteurs SURF associés aux points obtenus de cette manière. De cette façon, différents points de vue de l'objet sont enregistrés et ceci permet de pallier la sensibilité du descripteur SURF aux transformations affines.

Les images de la deuxième séquence sont utilisées pour construire la carte topologique de l'environnement. Nous selectionnons en réalité uniquement certaines images clefs obtenues par l'algorithme de SLAM métrique utilisé pour l'extraction automatique des objets. A chaque image clef est associé un lieu. Le modèle d'apparence est l'ensemble des descripteurs SURF extraits dans l'image clef auquel on a retiré les descripteurs appartenant aux objets dynamiques.

Un vocabulaire de mots visuels est appris avec la collection des images de l'environnement que nous avons à disposition. Les descripteurs SURF des modèles d'apparence sont quantifiés dans ce vocabulaire.

5.4 Calcul de vraisemblance du lieu et des objets présents

L'approche que nous avons retenue repose sur le modèle de sacs de mots visuels. Le calcul de vraisemblance est le calcul de similarité classique entre deux sacs de mots visuels :

$$s(h_1, h_2) = \frac{h_1^T h_2}{\|h_1\|_2 \|h_2\|_2} \tag{5.11}$$

h_1 et h_2 sont deux sacs de mots visuels. Les composantes de chaque vecteur sont pondérées par le terme de fréquence inverse 5.12 qui mesure l'importance du mot visuel dans l'environnement. Pour un mot visuel w, le terme de fréquence inverse

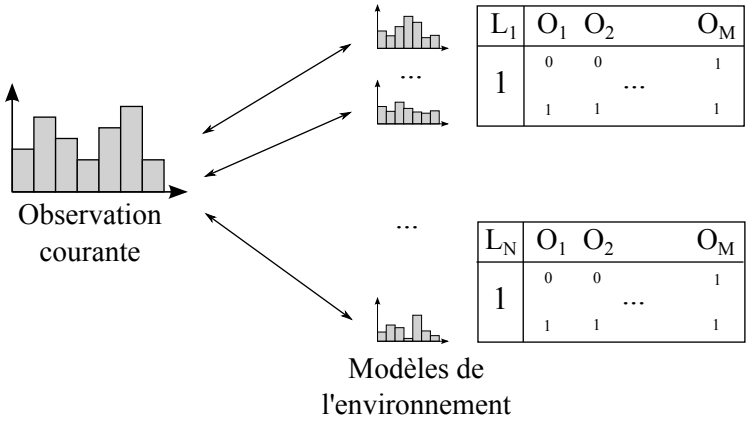


FIGURE 5.3 – **Classification de l'image courante.** Le sac de mots de l'image courante est comparé aux sacs de mots obtenus en faisant l'hypothèse d'un lieu et l'hypothèse de présence d'un ou plusieurs objets.

est donné par l'équation suivante (rappel du chapitre 2) :

$$IDF_w = ln\frac{N}{N_w} \tag{5.12}$$

avec N le nombre d'images dans la base de données et N_w le nombre d'images dans la base contenant le mot w.

Nous disposons d'une nouvelle séquence vidéo de l'environnement. Pour chaque image, nous souhaitons déterminer le lieu de prise de vue et les objets reconnus. Il faut donc calculer, pour chaque lieu, le score de similarité avec l'observation courante en supposant la présence d'un ou plusieurs objets. A l'hypothèse du lieu observé s'ajoute les hypothèses de présences de chaque objet. Le nombre d'hypothèses est donc multiplié par $2^{N_{objets}}$, avec N_{objets} le nombre d'objets dynamiques connus et enregistrés dans la base de données. Pour limiter les calculs, nous calculons dans un premier temps les scores de similarité de l'image courante avec tous les objets présents dans la base. Nous retenons les n_{objets} objets correspondant aux n_{objets} meilleurs scores obtenus. Nous estimons dans un second temps l'adéquation de l'image courante avec chaque hypothèse de lieu combiné avec l'hypothèse de présence d'un ou plusieurs objets parmi les n_{objets} retenus par l'étape préalable. Pour cela, le vecteur de mots visuels est construit à partir du modèle du lieu en ajoutant les mots visuels des objets qu'on suppose présents et la similarité est évaluée avec l'équation 5.11. On calcule les scores suivants :

$$s\left(h_{Observation}, \left\{h_{lieu} + \sum_{i=1}^{n_{objets}} (O_i == 1).h_{O_i}\right\}\right) \tag{5.13}$$

avec $h_{Observation}$ le vecteur de mots visuels de l'observation courante, h_{lieu} celui du lieu, h_{O_i} celui de l'objet i et O_i la variable binaire indiquant la présence de l'objet i. Dans les expériences $n_{objets} = 4$. La composition - lieu et objets présents - retenue est celle correspondant au score de similarité le plus élevé. La figure 5.3 illustre le procédé.

Une étape de validation géométrique par un calcul de matrice fondamentale rejette les fausses reconnaissances de lieu.

5.5 Expérimentations sur données réelles

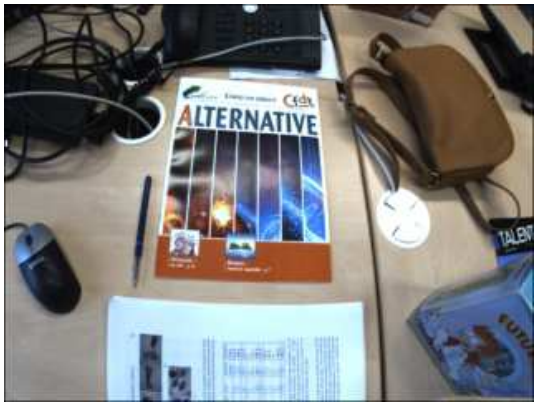
Nous évaluons les performances de notre algorithme sur les données *Bureaux* qui sont les données utilisées dans la troisième expérience du chapitre 3. Il s'agit de trois séquences vidéo prises à l'intérieur d'un bâtiment. Dans chaque séquence, la caméra explore les deux mêmes bureaux. 15 objets ont été déplacés et sont détectés par notre algorithme d'extraction automatique d'objet par la comparaison des séquences n° 1 et n° 2. Les images de la séquence n° 2 sont utilisées pour apprendre les modèles d'apparence des objets et la carte topologique de la structure statique de l'environnement. Nous exécutons notre nouvel algorithme de reconnaissance de lieu sur la séquence n° 3 et nous comparons les performances avec un algorithme de reconnaissance de lieu classique qui ne distingue pas les points de la structure statique et les points des objets. L'algorithme classique repose de la même manière sur l'approche en sacs de mots visuels, nous utilisons le même vocabulaire et une étape de validation géométrique avec des critères identiques à notre méthode.

La figure 5.5 présente un cas de vraisemblance ambigüe qui est due à la présence d'un objet dynamique dans l'image. On constate quatre pics sur la figure 5.5 (a). L'image courante est représentée sur la figure 5.4 (a). Les images correspondant aux deux pics les plus élevés sont représentés figure 5.4 (b) et (c). Les deux autres pics correspondent à un autre point de vue du même lieu (indice #286) présent dans la base de données à cause d'une boucle dans la trajectoire de la caméra, ainsi qu'à une vue d'apparence similaire indice #170 (*aliasing* perceptuel). Il est alors difficile de distinguer le lieu de provenance de l'image en considérant un environnement statique. Le calcul de vraisemblance que nous proposons permet de lever cette ambiguité. La valeur du pic correspondant à l'image 5.4 (c) (indice #420) est abaissée. Il reste le pic correspondant effectivement au lieu de provenance de l'image (indice #476).

5.5.1 Performances

Les tableaux 5.2 et 5.3 recensent les informations de l'expérience telle que le nombre d'images de la séquence (#Img), le nombre d'images clefs traitées par l'algorithme (#Img clefs), le nombre de reconnaissances de lieu correctes (# Reco. lieu), le nombre de reconnaissances de lieu incorrectes (# Reco. lieu err.) et le nombre d'objets correctement détectés (# Obj. detect.). On observe un nombre élevé de fausses détections avec l'algorithme classique. Elles sont toutes dues à la présence d'un objet déplacé. Le nombre de fausses détections diminue fortement avec notre algorithme mais il n'est pas nul. Ceci est dû à la difficulté de construire le modèle d'apparence de la structure statique. En pratique, la plupart des points appartenant aux objets dynamiques sont retirés du modèle de la structure statique. Cependant, certains points restent associés à la structure statique par erreur et perturbent encore la reconnaissance du lieu. La diminution du taux d'erreurs montre néanmoins l'intérêt d'une telle méthode.

Le deuxième intérêt de notre méthode est la reconnaissance des objets en parallèle de la reconnaissance de lieu. Notre algorithme parvient en effet à détecter les 15 objets qui sont effectivement observés dans la séquence.



(a)

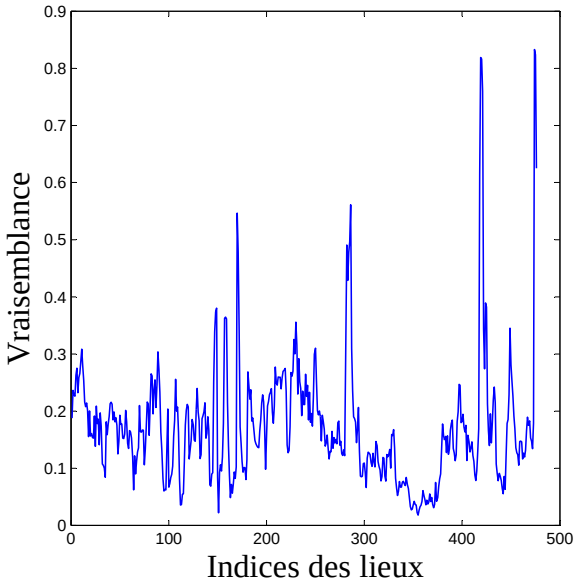
(b)

(c)

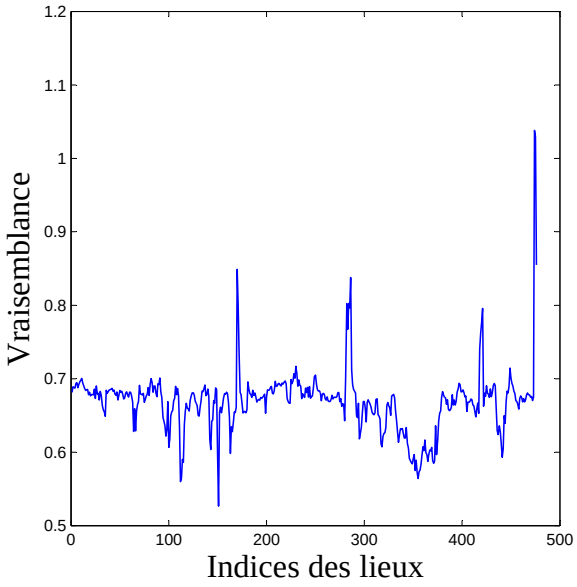
FIGURE 5.4 – **Données menant à un calcul de vraisemblance ambigüe.** (a) Image courante. (b)Image #476, (c) Image #420. Les images d'indices #476 et #420 proviennent de deux lieux différents

#Img	#Img clefs	# Reco. lieu	# Reco. lieu err.	# Obj. detect.
2448	500	310	**83**	**0**

TABLE 5.2 – Performance de l'algorithme de reconnaissance classique



(a)



(b)

FIGURE 5.5 – **Un exemple de vraisemblance ambigüe.** (a) Vraisemblance de l'image avec un calcul classique. (b) Vraisemblance calculée avec notre méthode. L'ambiguité est levée avec notre algorithme.

5.6 Discussions

Dans ce chapitre, nous avons avant tout souhaité montrer qu'une connaissance plus fine de l'environnement peut permettre d'améliorer une méthode de SLAM to-

#Img	#Img clefs	# Reco. lieu	# Reco. lieu err.	# Obj. detect.
2448	500	267	**25**	**15**

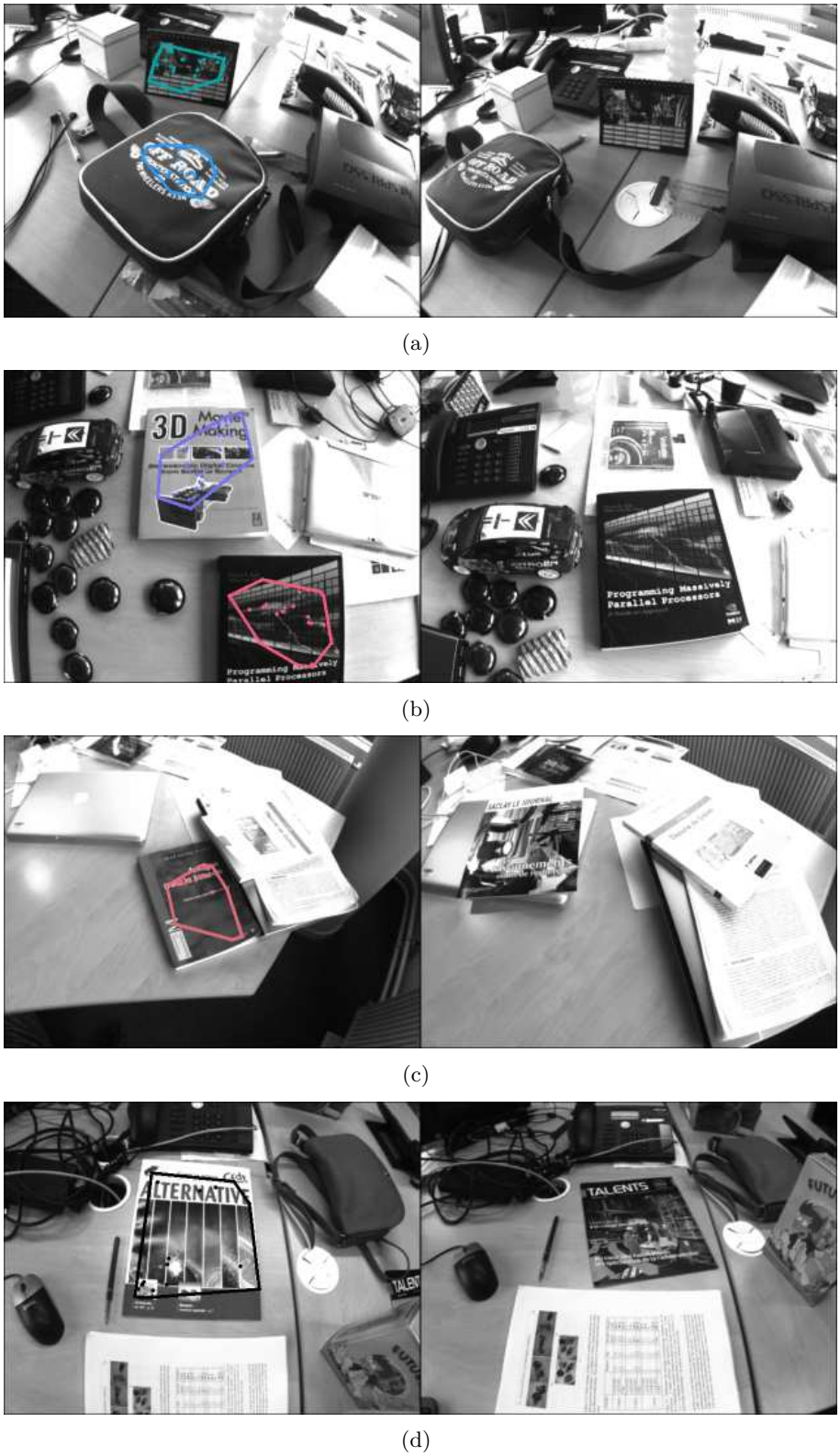
TABLE 5.3 – Performance de l'algorithme de reconnaissance de lieu prenant en compte la présence des objets dynamiques

pologique. La méthode présentée ici permet de se localiser dans une carte topologique et, en parallèle, reconnaître des objets appris au fur et à mesure de l'exploration de la caméra dans l'environnement (figure 5.6). Notre méthode considère que l'environnement est dynamique et peut être ainsi comparée favorablement avec l'ensemble des méthodes citées dans l'état de l'art.

Nous avons mentionné trois solutions au problème du SLAM topologique dans l'état de l'art [Fraundorfer *et al.*, 2007, Cummins et Newman, 2010, Angeli *et al.*, 2008]. Ces solutions font l'hypothèse d'un environnement statique et déterminent le lieu de provenance d'une image en s'appuyant sur le concept de sacs de mots visuels. Elles utilisent aussi une étape supplémentaire de validation géométrique indispensable au bon fonctionnement du système puisque les algorithmes présentés sont incrémentaux. Elle se placent donc dans un cadre proche du nôtre. Nous avons montré que ce type d'algorithme était perturbé par la présence d'objets dynamiques. Avec notre nouvelle formulation, le nombre d'erreurs de reconnaissance de lieu est moindre.

Certaines améliorations peuvent néanmoins être apportées. Tout d'abord, lors de la construction de la carte topologique, il est difficile de bien séparer les points des objets et les points de la structure statique. Pour séparer l'ensemble des pixels de l'image appartenant aux objets avant la mise à jour du modèle d'apparence du lieu, nous pourrions utiliser une méthode dense ou combiner notre solution d'extraction d'objet avec une méthode de segmentation. D'autre part, les objets sont souvent des parties saillantes de l'environnement qui concentrent un grand nombre de points d'intérêt. En retirant les points des objets du modèle d'apparence de la structure statique, les modèles des lieux, basés sur les descripteurs de points d'intérêt, sont de moins en moins discriminants. Il est alors difficile de distinguer le lieu de provenance de l'image de manière robuste. L'utilisation d'autres types de primitives, comme les segments ou les lignes, serait judicieuse.

Par ailleurs, nous montrons dans ce chapitre comment améliorer une méthode de SLAM topologique, la modélisation que nous proposons peut aussi être appliquée à une méthode de SLAM métrique. Si nous considérons que tous les objets potentiellement dynamiques ont été appris et enregistrés dans une base de données, nous pouvons recherchés les objets dynamiques dans l'image courante et exclure les points de ces objets du calcul de pose. Lorsque la scène est dynamique, le calcul est plus robuste puisqu'il repose uniquement sur les points de la structure statique de la scène.



(a)

(b)

(c)

(d)

FIGURE 5.6 – **Resultats obtenus avec notre méthode de localisation topologique.** A gauche, l'image courante et les objets détectés. A droite, l'image correspondant au lieu de provenance enregistré dans la carte topologique

Conclusion

Travaux réalisés

Dans cette thèse, nous nous sommes attachés à proposer une solution pour modéliser une scène intérieure dynamique sans aucune connaissance *a priori* à partir des images d'une caméra monoculaire à champ moyen. Nous sommes parti du constat que les changements survenus dans un environnement intérieur sont en grande partie dus aux mouvements d'objets saillants. Notre objectif est de détecter tous les changements survenus dans une scène 3D en comparant deux explorations d'une caméra dans l'environnement. Une fois les mouvements détectés, nous pouvons segmenter la partie statique des structures dynamiques et fournir une description de la scène exploitable et aisément actualisable. Pour décrire les environnements dynamiques, nous proposons une nouvelle définition de la scène qui consiste en une structure statique et un ensemble d'objets dynamiques. Un objet est une entité rigide qu'un utilisateur peut prendre et déplacer et que l'on peut repérer visuellement.

La modélisation présentée dans ces travaux ne dispose d'aucune carte de l'environnement ni d'information préalable sur les objets présents. Nous apprenons automatiquement les objets de la scène en analysant et comparant plusieurs explorations d'une caméra monoculaire dans un environnement. Contrairement aux approches de SLAM existantes qui ignorent les incohérences dues aux mouvement d'objets dynamiques qu'elles traitent comme une aberration, notre approche tire partie de ces incohérences pour apprendre de nouveaux objets. Cette source d'information supplémentaire permet une connaissance plus approfondie de la scène avec l'enrichissement d'une base de données d'objets au fur et à mesure des explorations.

L'approche proposée est l'extraction d'objet par le mouvement : un objet est défini comme un ensemble de points ayant un mouvement cohérent par rapport à la structure statique de la scène. Pour cela, nous avons dans un premier temps associé une méthode de reconnaissance de lieu reposant sur l'apparence à une méthode de SLAM métrique pour pouvoir se relocaliser précisément dans une carte éparse des points 3D de la scène. La méthode permet de déterminer la structure statique de la scène et de détecter des incohérences dans la carte métrique et dans la carte topologique de la scène. Elle fournit aussi deux vues d'un même lieu prises à des instants différents. L'analyse de ces deux vues permet d'expliquer les incohérences observées. Un nouvel algorithme de détection de multiples structures entre deux vues a été développé pour répondre à ce problème. L'association de ces deux contributions

permet de construire pour chaque objet un modèle d'apparence et un modèle 3D. De cette manière, nous pouvons maintenir et actualiser le modèle de la scène à chaque nouvelle exploration. D'autre part, au fur et à mesure de l'évolution de l'environnement, nous enrichissons la base de données d'objets susceptibles d'être présents dans la scène.

Enfin, nous avons appliqué notre modélisation à l'amélioration d'une méthode de reconnaissance de lieu. Chaque lieu est modélisé par les mots visuels appartenant à la structure statique de la scène. L'apparence des objets dynamiques est apprise avec notre méthode d'extraction par le mouvement sur des explorations précédentes. La modélisation de la scène que nous proposons permet de faire l'hypothèse de se trouver dans un lieu et de supposer la présence d'un ou plusieurs objets. On détermine alors le lieu associé à l'observation et les objets présents par un critère de maximum de vraisemblance. Cette méthode permet de rejeter un grand nombre de fausses reconnaissances dues à la présence d'objets ayant bougé.

Les études reportées dans ce mémoire ont montré la possibilité de modéliser un environnement dynamique et d'exploiter la modélisation dans une application de modélisation en utilisant uniquement les données visuelles provenant d'une caméra monoculaire. De cette façon, il est possible de décrire une scène et son évolution au cours du temps.

Perspectives

L'objectif des travaux de cette thèse a été de montrer l'intérêt de définir une scène par une structure statique et un ensemble d'objets dynamiques pour modéliser un environnement et son évolution au cours du temps. Nos travaux sont avant tout des démonstrations de la faisabilité de cette idée. Les expériences ont mis en évidence des résultats encourageants et prouvent l'intérêt de l'approche proposée. Elles montrent aussi que certaines caractéristiques pourraient être améliorées pour augmenter l'efficacité de notre solution.

Détermination de la structure statique

Les différentes expériences ont montré qu'il était parfois difficile de déterminer de façon précise la structure statique de la scène, notamment lorsqu'une grande partie de la scène est modifiée par le mouvement des objets. Ceci est dû au fait que notre solution repose sur la reconnaissance de points d'intérêt. Les objets saillants contiennent une grosse proportion des points observés. La structure statique est souvent constituée des murs et du mobilier permanent et est difficile à décrire avec un ensemble de points. L'utilisation d'autres types de primitives comme les droites ou les segments apporterait des informations supplémentaires et permettrait de mieux caractériser la structure statique.

Caractérisation d'un objet disparu

Pour qu'un objet soit caractérisé par notre solution, il faut qu'il soit déplacé au sein d'un lieu ou bien qu'il soit observé dans deux lieux différents. Il serait intéressant de pouvoir caractériser un objet directement en détectant sa disparition d'un lieu. Pour cela, nous devons comparer des vues d'un lieu contenant l'objet et des vues du même lieu ne contenant plus l'objet. L'utilisation d'une carte de points 3D éparse rend difficile la détermination d'un objet disparu. Un algorithme de SLAM dense permettrait de caractériser plus précisément un ensemble de points disparus et de le définir comme un objet.

Performance temps réel

L'implémentation de l'algorithme de détection de multiples structures entre deux vues telle qu'elle existe ne permet pas de fonctionner en temps réel. Cette partie, hautement parallélisable, pourrait néanmoins bénificier des avancées récentes sur le calcul embarqué massivement parallèle (implémentation sur GPU par exemple).

Annexe A

Principes de base

A.1 La caméra perspective

A.1.1 Le modèle sténopé

Les capteurs utilisés pour l'acquisition d'images ou de séquences vidéos sont décrits par le modèle sténopé (*pinhole* en anglais) qui permet une modélisation simple du processus de formation des images au sein d'une caméra. Il est aussi appelé le *modèle de projection perspective* ou *modèle de projection centrale*. La caméra est modélisée par une projection centrale : tous les rayons de lumière passent par un seul et même point appelé le centre optique. L'image se forme dans le plan image naturellement situé derrière le centre optique. La projection perspective créé une image inversée. On considère donc souvent l'image virtuelle associée au plan placé devant le centre optique à la même distance que le plan image.

A.1.2 La projection perspective d'un point 3D

L'image fournie par une caméra est formée de l'ensemble des projections sur le plan image des points 3D de la scène situés dans le champ de vue. Un point 3D visible $\mathbf{M}$ de $\mathbb{R}^3$ est projeté sur le plan image en un point m. C'est l'intersection du plan de la rétine de la caméra (*i.e* capteur) avec le rayon de projection de $\mathbf{M}$ dirigé vers le centre optique. La projection peut se décomposer en trois transformations élémentaires successives (voir figure A.1) :

1. Le changement de repère qui consiste à exprimer les coordonnées de $\mathbf{P}$ dans le repère lié à la caméra. Cette transformation définit les **paramètres extrinsèques** de la caméra.
2. La transformation entre le repère caméra et le repère capteur. C'est la projection centrale du point 3D dans le plan de la rétine.
3. La transformation entre le repère capteur (repère lié à la physique du capteur et où les coordonnées sont exprimées en mm) et le repère image (repère

géométrique où les coordonnées sont exprimées en pixels).

Les deux dernières transformations définissent les **paramètres intrinsèques** de la caméra.

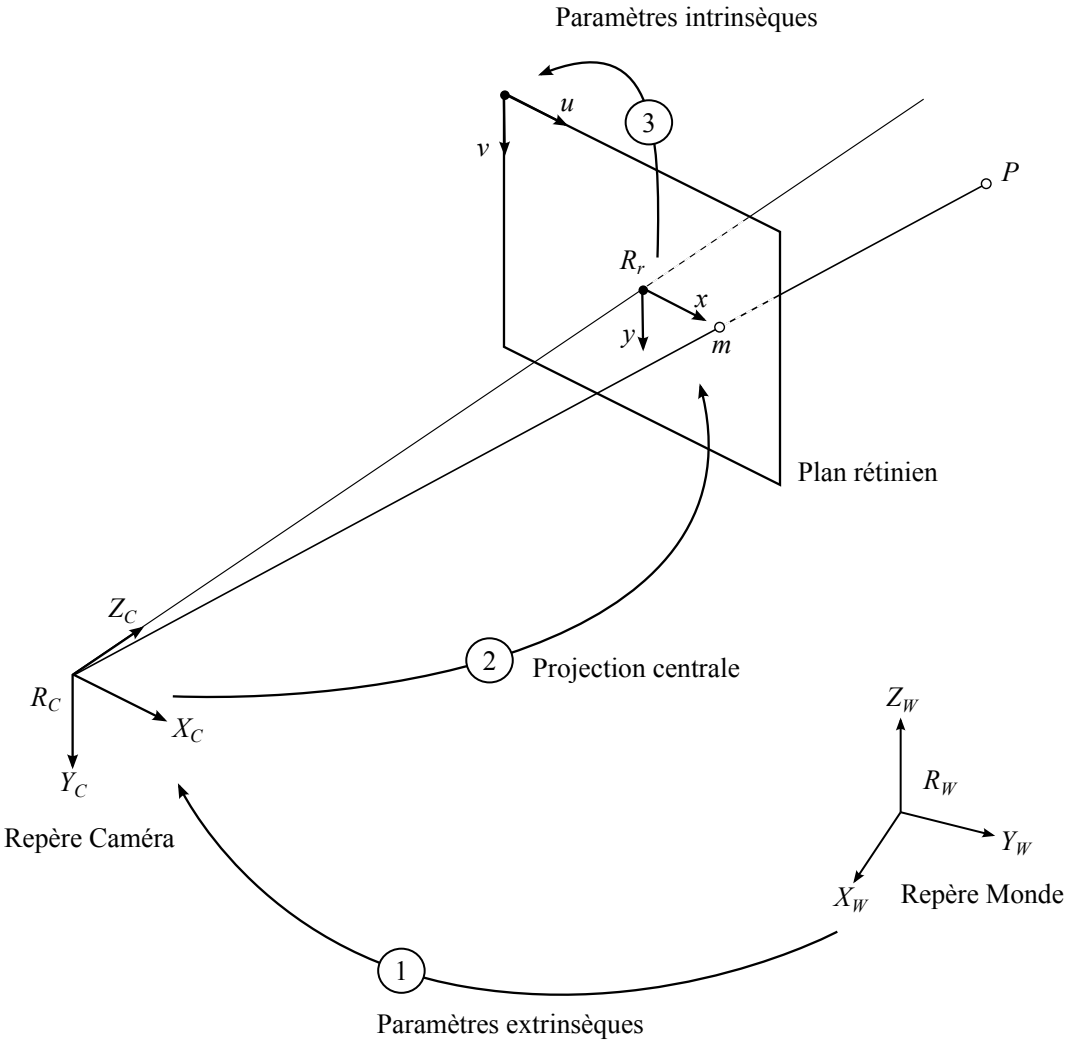


FIGURE A.1 – **Projection perspective.** La projection perspective peut être vue comme trois transformations géométriques consécutives.

Les points sont représentés par leurs coordonnées homogènes. Un point 2D de coordonnées euclidiennes (x, y) est représenté par le vecteur $(x, y, 1)^T$. De la même façon, un point 3D de coordonnées euclidiennes (x, y, z) est représenté par le vecteur $(x, y, z, 1)^T$.

A.1.3 Paramètres extrinsèques

La première étape de la projection est le changement de repère entre le repère du monde $\mathcal{R}_\mathcal{W}$ et le repère lié à la caméra $\mathcal{R}_\mathcal{C}$. Cette transformation rigide peut se décomposer en une rotation $\mathbf{R}$ et une translation $\mathbf{t}$. Les points 3D sont indicés $\mathcal{W}$ ou $\mathcal{C}$ en fonction du repère dans lequel leurs coordonnées sont exprimées, respectivement

le repère monde et le repère caméra. On peut écrire les relations suivantes :

$$\mathbf{M}_{\mathcal{C}} = \begin{bmatrix} X_{\mathcal{C}} \\ Y_{\mathcal{C}} \\ Z_{\mathcal{C}} \\ 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_{\mathcal{W}} \\ Y_{\mathcal{W}} \\ Z_{\mathcal{W}} \\ 1 \end{bmatrix} = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M}_{\mathcal{W}} \tag{A.1}$$

et

$$\mathbf{M}_{\mathcal{W}} = \begin{bmatrix} R^T & -R^T\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M}_{\mathcal{C}} \tag{A.2}$$

Les paramètres de cette transformation correspondent aux **paramètres extrinsèques** de la caméra qui caractérisent la pose de celle-ci dans le repère monde. La pose de la caméra possède six degrés de liberté :

- La position du centre optique, décrit par le vecteur $\mathbf{t} = (t_x, t_y, t_z)^T$
- L'orientation de la caméra décrit par la matrice de rotation R.

A.1.4 Paramètres intrinsèques

La deuxième transformation est une projection perspective qui transforme le point 3D $(X_{\mathcal{C}}, Y_{\mathcal{C}}, Z_{\mathcal{C}})$ en un point-image (x, y). Cette transformation relie le repère caméra $\mathcal{R}_{\mathcal{C}}$ avec le repère associé au plan rétinien $\mathcal{R}_r$.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_{\mathcal{C}} \\ Y_{\mathcal{C}} \\ Z_{\mathcal{C}} \\ 1 \end{bmatrix} \tag{A.3}$$

f est la *distance focale* de l'objectif utilisé. Elle décrit la distance orthogonale entre le centre optique et le plan de la rétine de la caméra.

La troisième transformation décrit l'opération de conversion des coordonnées en unité métrique (x, y) en coordonnées en pixels (u,v).

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_x & 0 & u_0 \\ 0 & k_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{A.4}$$

où :

- u_0 et v_0 désignent les coordonnées en pixels de l'intersection de l'axe optique avec le plan de la rétine. Ce sont les coordonnées du *point principal*
- k_x et k_y désignent le nombre de pixels par unité de longueur suivant les directions x et y du capteur.

Les paramètres (f, k_x, k_y, u_0, v_0) sont les **paramètres intrinsèques** de la caméra. Ils sont obtenus par une étape de *calibration* de la caméra.

A.1.5 Paramètres de distorsion

Le modèle de projection perspective est un modèle idéal qui approxime le modèle optique des lentilles minces. En réalité, il n'est pas assez précis pour les sys-

tèmes de vision utilisés, notamment lorsque l'objectif est un objectif grand angle à courte focale. L'optique introduit des distorsions qui se manifestent par des déformations dans l'image : les lignes droites deviennent des courbes. Elle est prise en compte dans le modèle pour obtenir la relation entre les coordonnées du points 3D et les coordonnées du point image déformé par la distorsion. La distorsion possède une composante tangentielle et une composante radiale. La distorsion tangentielle, beaucoup plus faible, est négligée. On modélise la distorsion radiale par une fonction reliant les coordonnées « idéales » du point dans l'image $m = (x, y)$ aux coordonnées « réelles » mesurées $\breve{m} = (\breve{x}, \breve{y})$:

$$\begin{bmatrix} \breve{x} \\ \breve{y} \end{bmatrix} = L(r) \begin{bmatrix} x \\ y \end{bmatrix} \tag{A.5}$$

avec :
- (x, y) les coordonnées idéales du point dans l'image
- $(\breve{x}, \breve{y})$ les coordonnées réelles du point dans l'image, avec les effets de la distorsion
- r la distance radiale $\sqrt{x^2 + y^2}$
- $L(r)$ le facteur de distorsion qui dépend uniquement de la distance radiale. $L(r)$ est décrit par un polynôme de degré 5.

Nous considérons par la suite que le capteur utilisé est calibré et que les paramètres de distorsion sont connus. Les images sont corrigées en distorsion pour obtenir les données acquises par un système idéal.

A.2 Géométrie entre deux vues

A.2.1 Géométrie épipolaire

La géométrie épipolaire est un modèle mathématique qui décrit les relations géométriques entre deux caméras observant la même scène rigide. Dans le cas où les paramètres internes des caméras sont inconnus, elle est exprimée par la matrice fondamentale F, qui est une matrice 3×3 de rang 2. Si un point 3D P se projette sur le point p_1 dans la première vue et sur le point p_2 dans la deuxième, les points vérifient la relation :

$${p_2}^T F p_1 = 0. \tag{A.6}$$

Cette relation permet d'estimer la matrice fondamentale à partir d'un ensemble de correspondances de points 2D. En pratique, F se calcule avec l'algorithme des 8 points [Hartley, 1997].

Nous quantifions la distance d'une association de points 2D (p_1, p_2) à une matrice F par la distance de Sampson [Hartley et Zisserman, 2000], qui est une approximation au premier ordre de l'erreur géométrique de reprojection :

$$e_{Sampson} = \frac{\left({p_2}^T F p_1\right)^2}{(Fp_1)_1^2 + (Fp_1)_2^2 + (F^T p_2)_1^2 + (F^T p_2)_2^2} \tag{A.7}$$

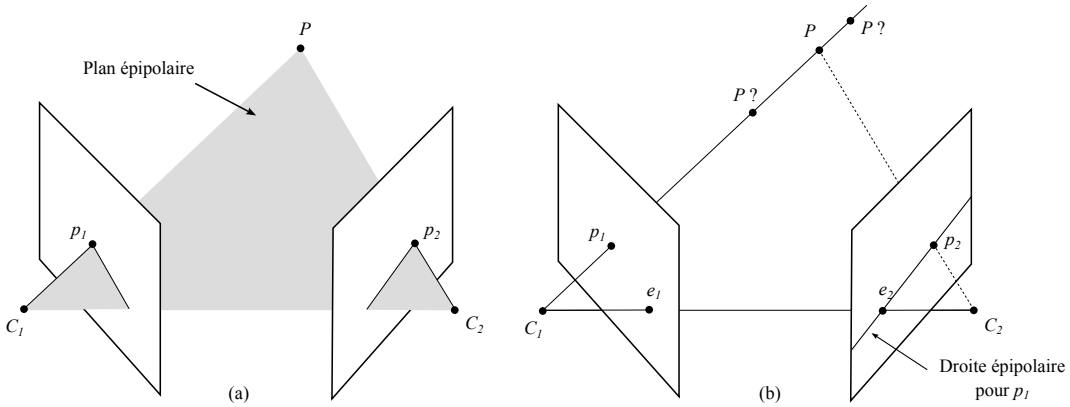


FIGURE A.2 – **Géométrie épipolaire.** (a) Les deux caméras sont représentées par leurs centres de projection C_1 et C_2 et les plans image correspondants. Ces centres de projection, le point P et ses projections p_1 et p_2 dans les deux plans image appartiennent à un même plan. (b) On calcule, pour le point p_1 de la première vue, la droite épipolaire $l_2 \sim Fp_1$ sur laquelle se situe l'observation correspondante dans la deuxième vue. Le point image p_2 est l'intersection de cette droite avec la droite (PC_2). Dans chacune des images, un point joue un rôle particulier. Il s'agit des deux épipôles e_1 et e_2. Ils correspondent à la projection dans l'image du centre optique de l'autre caméra et au point d'intersection de toutes les droites épipolaires de l'image.

En effet, en pratique la matrice doit être estimée à partir d'un grand nombre de correspondances, le problème est surdéterminé et les données sont bruitées. Il est essentiel de connaître l'erreur résiduelle ou résidu d'une correspondance par rapport au modèle recherché. La distance de Sampson est communément utilisée en vision par ordinateur pour estimer de façon précise ce résidu.

A.2.2 Homographie 2D

Lorsque la scène observée est planaire, les relations géométriques sont décrites par une homographie 2D. Une homographie (ou application projective) est une transformation linéaire bijective entre deux espaces projectifs qui préserve l'alignement. Etant donné deux points image p_1 et p_2, projection du même point 3D P (voir figure A.3), ces points sont reliés par la matrice H de taille 3×3 de la manière suivante :

$$p_2 \sim Hp_1 \tag{A.8}$$

La matrice H est définie à un facteur près, elle a 8 degrés de liberté, elle peut être déterminée avec 4 correspondances de points.

De la même façon que pour la matrice fondamentale, on utilise l'erreur de Sampson pour mesurer la distance d'une association de points image 2D à une homographie. La relation A.8 peut être écrite sous la forme du système d'équations $Ah = 0$, h étant le vecteur contenant les 9 composantes de la matrice H. La distance de Sampson pour une homographie s'écrit alors :

$$e^2_{Sampson} = h^T A^T \left(JJ^T\right)^{-1} Ah, \tag{A.9}$$

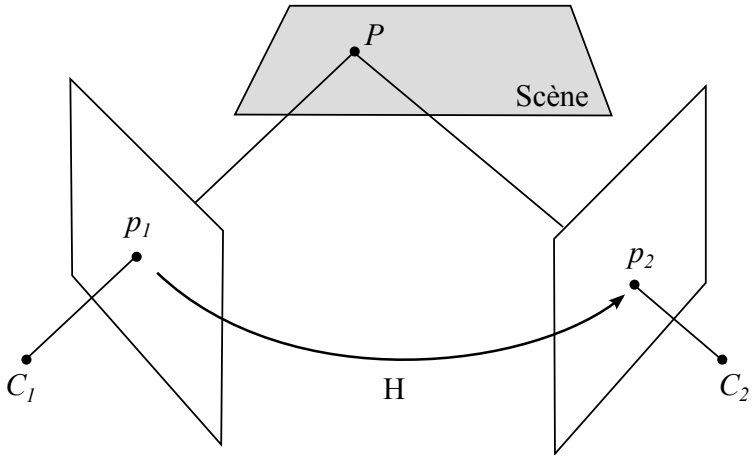


FIGURE A.3 – **Homographies 2D.** Les coordonnées des observations correspondantes de points 3D situés sur un même plan de l'espace sont reliées par une homographie 2D.

où $J = \frac{\partial(Ah)}{\partial(\tilde{p})}$ est la jacobienne du système d'équations, $\tilde{p}$ les coordonnées de deux points appariés.

A.3 Géométrie de l'environnement

A.3.1 Calcul de la structure de l'environnement par triangulation

La triangulation est la technique permettant de déterminer la position d'un point 3D à partir de ces observations dans au moins deux caméras dont on connait les poses et le calibrage. Connaissant la pose d'une caméra et l'observation d'un point 3D dans l'image, il est possible de rétroprojeter le point pour inférer la position 3D. La rétroprojection peut être vue comme l'opération inverse de la projection. C'est le rayon optique passant par le centre de la caméra et par l'observation dans l'image. Il n'est donc pas possible d'avoir la profondeur du point à partir d'une seule image. Si le point 3D est observé dans deux caméras, sa position est déterminée par triangulation à l'intersection des rayons optiques issus des deux observations (figure A.4). En présence de bruit dans les données (dû à la précision du calibrage, des poses calculées et des positions des points détectés), les rayons 3D ne sont pas forcément sécants. Dans ce cas, le résultat de la triangulation du point 3D est le point équidistant aux deux rayons. C'est le point Q sur la figure A.4. Cette méthode rapide peut être étendue à plus de deux vues pour plus de robustesse. Par exemple, dans le cas de 3 caméras, il est possible de calculer 3 triangulations différentes à partir des couples de caméras. Le résultat final de la triangulation est le barycentre de ces 3 points. Cette position peut être raffinée par optimisation non linéaire, par exemple dans la méthode de SLAM métrique de MOuragnon et al. [Mouragnon *et al.*, 2006].

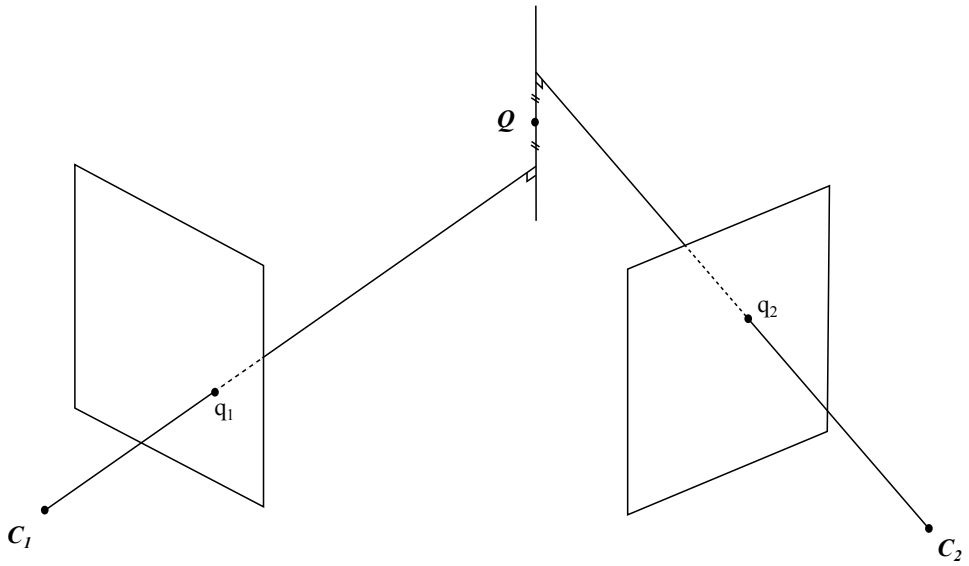


FIGURE A.4 – **Triangulation de points 3D.** La structure de l'environnement peut être obtenue par triangulation des observations dans les images.

A.3.2 Calcul de la pose de la caméra

Une fois que l'environnement a été partiellement reconstruit, il est possible de calculer la pose d'une caméra à partir de correspondances entre les points 2D et des points 3D de position connues. On cherche alors à localiser la caméra dans le repère défini par ces points 3D. Le calcul de pose nécessite de connaître la position d'au moins 3 points 3D dans le cas de caméra calibrée et 6 dans le cas non calibré. De nombreuses méthodes ont été proposées pour résoudre ce problème. On peut trouver une comparaison des méthodes de base dans [Haralick *et al.*, 1994]. Plus récemment, les auteurs de [Lepetit *et al.*, 2009] ont présenté une méthode plus rapide et plus précise.

Il est à noter que les poses de caméras peuvent être aussi retrouvées à partir des associations 2D/2D en déterminant la matrice fondamentale de la transformation entre les deux vues. La matrice fondamentale est en effet reliée aux poses de la caméra par la relation A.10.

$$F = [e_2]_\times * P_2 * P_1^{-1} \tag{A.10}$$

où P_1 et P_2 sont les matrices de pose des caméras respectivement liées à la première et à la deuxième image. $e_2 = P_2 * C_1$ avec C_1 le centre de la première caméra. On note $\times$ le produit vectoriel. La matrice $[x]_\times$ est la matrice de produit vectoriel. Elle est définie telle que pour tout vecteur à 3 composantes y, on a $[x]_\times y = x \times y$:

$$[x]_\times = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{A.11}$$

Une fois la matrice fondamentale déterminée à partir des correspondances de points, les caméras peuvent être choisies de la manière suivante :

$$P_1 = [I|\mathbf{0}] \text{ et } P_2 = \left[[\mathbf{e_2}]_\times F|\mathbf{e_2}\right] \tag{A.12}$$

où $\mathbf{e_2}$ est l'épipole vérifiant $\mathbf{e_2}^T F = 0$.

L'utilisation d'associations 2D/3D plutôt que 2D/2D présente plusieurs avantages. Le calcul de pose 2D/3D est beaucoup plus rapide que le calcul de pose 2D/2D. L'estimation de la matrice fondamentale est en effet difficile et coûteuse en temps de calcul. De plus, l'extraction des paramètres à partir de la matrice fondamentale ne permet pas d'estimer le facteur d'échelle et donc en particulier la norme de la translation entre les différentes caméras. Avec l'approche 2D/3D, le facteur d'échelle est estimé à partir de l'observation de la distance entre les différents points de l'espace. Enfin, les auteurs de [Tardif *et al.*, 2008] ont montré que l'utilisation de l'approche 2D/3D offre un calcul plus précis de la position de la caméra.

De la même façon que pour l'estimation d'une matrice fondamentale ou d'une homographie, les algorithmes cités dans cette section estiment la pose avec un échantillon de taille minimale. Pour être robuste aux données aberrantes, ils sont conjointement utilisés avec des techniques d'estimation robuste comme la procédure RANSAC.

A.3.3 Ajustement de faisceaux

L'ajustement de faisceaux est une méthode d'optimisation qui raffine une première estimation des paramètres d'une scène 3D à l'aide d'un algorithme d'optimisation non linéaire, comme l'algorithme de Levenberg-Marquardt.

Lorsqu'un ensemble de points 3D et de caméras sont reconstruits avec les méthodes définies précédemment, il est nécessaire de mesurer la qualité de la reconstruction en définissant une erreur. Cette erreur caractérise l'adéquation entre le modèle reconstruit et les observations du modèle. Pour chaque point reconstruit, l'erreur doit mesurer la distance entre l'endroit où le point est détecté dans l'image et sa position estimée. La solution retenue dans ces travaux est l'erreur de reprojection géométrique (figure A.5) qui mesure la distance 2D entre l'observation du point 3D dans l'image (repérée par le point d'intérêt 2D extrait dans l'image) et la projection du point 3D reconstruit dans cette image :

$$r = \|q - \pi(PQ)\| \tag{A.13}$$

L'ajustement de faisceaux raffine l'ensemble des paramètres de la scène 3D (constitué des 6 paramètres de pose de chaque caméra et des 3 paramètres de la position de chaque point 3D) en minimisant l'erreur de reprojection de chacun des couples caméra-point 3D observé. La fonction à minimiser s'écrit de la manière suivante :

$$F(C_1, ..., C_N, Q_1, ..., Q_M) = \sum_{1 \leq j \leq N} \sum_{i \in A_j} \left\| q_j^i - \pi(P_j Q^i) \right\| \tag{A.14}$$

$(C_i)_i$ sont les paramètres extrinsèques des caméras, $(P_i)_i$ sont les matrices de projection des caméras, $(Q_i)_i$ sont les points 3D et q_j^i est l'observation du point i dans la caméra j. L'ensemble A_j est l'ensemble des indices des points 3D vus par la caméra j. Cette fonction est minimisée par la méthode d'optimisation non-linéaire de Levenberg-Marquardt [Levenberg, 1944].

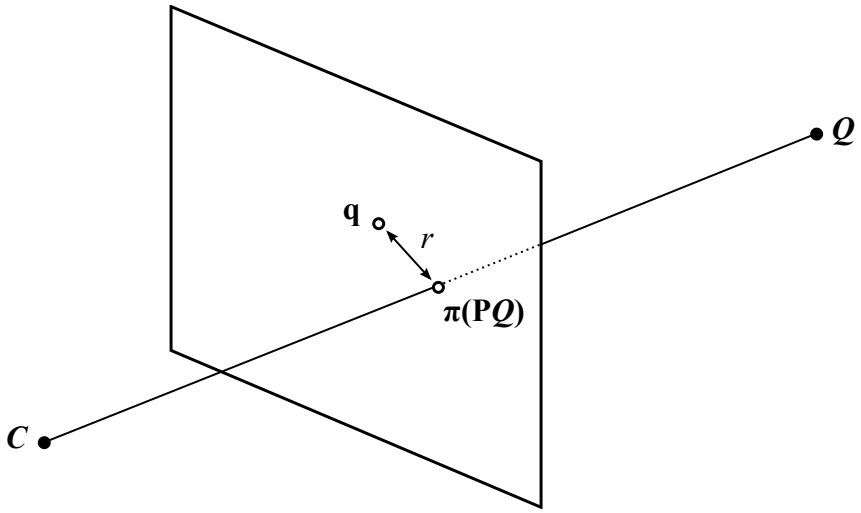


FIGURE A.5 – **Erreur de reprojection.** L'erreur de reprojection est la distance entre l'observation q d'un point Q et sa projection dans l'image $\pi(PQ)$.

Table des figures

Liste des tableaux

Bibliographie

[Alahi *et al.*, 2012] Alahi, A., Ortiz, R. et Vandergheynst, P. (2012). Freak : Fast retina keypoint. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517.

[Angeli et Davison, 2010] Angeli, A. et Davison, A. (2010). Live feature clustering in video using appearance and 3D geometry. *In Proceedings of the British Machine Vision Conference*, pages 41.1–41.11.

[Angeli *et al.*, 2008] Angeli, A., Filliat, D., Doncieux, S. et Meyer, J.-A. (2008). A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*.

[Bay *et al.*, 2006] Bay, H., Tuytelaars, T. et Gool, L. J. V. (2006). Surf : Speeded up robust features. *In Proceedings of the European Conference on Computer Vision*, pages 404–417.

[Beuter *et al.*, 2010] Beuter, N., Swadzba, A., Kummert, F. et Wachsmuth, S. (2010). Using articulated scene models for dynamic 3d scene analysis in vista spaces. *3D Research*, 1(3):20 :1–20 :13.

[Biswas *et al.*, 2002] Biswas, R., Limketkai, B., Sanner, S. et Thrun, S. (2002). Towards object mapping in dynamic environments with mobile robots. *In Proceedings of the Conference on Intelligent Robots and Systems (IROS)*.

[Botterill *et al.*, 2009] Botterill, T., Green, R. et Mills, S. (2009). A Bag-of-Words Speedometer for Single Camera SLAM. *In Image and Vision Computing New Zealand*, pages 1–6.

[Boykov *et al.*, 2001] Boykov, Y., Veksler, O. et Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[Bradski, 1998] Bradski, G. R. (1998). Computer vision face tracking for use in a perceptual user interface. *In Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*.

[Calonder *et al.*, 2010] Calonder, M., Lepetit, V., Strecha, C. et Fua, P. (2010). Brief : binary robust independent elementary features. *In Proceedings of the 11th European conference on Computer vision*, pages 778–792.

[Castle *et al.*, 2008] Castle, R. O., Klein, G. et Murray, D. W. (2008). Video-rate localization in multiple maps for wearable augmented reality. *In Proceedings of the 12th IEEE International Symposium on Wearable Computers*, pages 15–22.

[Castle *et al.*, 2010] Castle, R. O., Klein, G. et Murray, D. W. (2010). Combining monoslam with object recognition for scene augmentation using a wearable camera. *Image Vision Computing*, 28(11):1548–1556.

[Chenavier et Crowley, 1992] Chenavier, F. et Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry. *In Proceedings of the IEEE International Conference on Robotics and Automation*.

[Chin *et al.*, 2009] Chin, T.-J., Wang, H. et Suter, D. (2009). Robust fitting of multiple structures : The statistical learning approach. *In Proceedings of the IEEE International Conference on Computer Vision*, pages 413–420.

[Chin *et al.*, 2012] Chin, T.-J., Yu, J. et Suter, D. (2012). Accelerated hypothesis generation for multistructure data via preference analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):625–638.

[Chum et Matas, 2005] Chum, O. et Matas, J. (2005). Matching with prosac - progressive sample consensus. *In Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 220–226.

[Chum *et al.*, 2003] Chum, O., Matas, J. et Kittler, J. (2003). Locally optimized ransac. *In Proceedings of the 25th DAGM Symposium*, pages 236–243.

[Comaniciu et Meer, 1998] Comaniciu, D. et Meer, P. (1998). Distribution free decomposition of multivariate data. *Pattern Analysis and Applications*, 2:22–30.

[Comaniciu *et al.*, 2002] Comaniciu, D., Meer, P. et Member, S. (2002). Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:603–619.

[Comaniciu *et al.*, 2001] Comaniciu, D., Ramesh, V. et Meer, P. (2001). The variable bandwidth mean shift and data-driven scale selection. *In in Proceedings of the 8th International Conference on Computer Vision*, pages 438–445.

[Comaniciu *et al.*, 2003] Comaniciu, D., Ramesh, V. et Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575.

[Crowley, 1989] Crowley, J. L. (1989). World modeling and position estimation for a mobile robot using ultrasonic ranging. *In IEEE International Conference on Robotics and Automation*, pages 674–680.

[Crowley *et al.*, 1989] Crowley, J. L., Reignier, P. et Bobet, P. (1989). Dynamic modeling of free-space for a mobile robot. *In IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 623–633.

[Cucchiara *et al.*, 2003] Cucchiara, R., Grana, C., Piccardi, M. et Prati, A. (2003). Detecting moving objects, ghosts and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1337–1342.

[Cummins et Newman, 2007] Cummins, M. et Newman, P. (2007). Probabilistic Appearance Based Navigation and Loop Closing. *IEEE International Conference on Robotics and Automation*.

[Cummins et Newman, 2009] Cummins, M. et Newman, P. (2009). Highly scalable appearance-only SLAM : Fab-map 2.0. *In Proceedings of Robotics : Science and Systems*.

[Cummins et Newman, 2010] Cummins, M. et Newman, P. (2010). FAB-MAP : Appearance-Based Place Recognition and Mapping using a Learned Visual Vocabulary Model. *In Proceedings of the 27th International Conference on Machine Learning*.

[Davison, 2003] Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. *In Proceedings of the 9th IEEE International Conference on Computer Vision*.

[Davison *et al.*, 2007] Davison, A. J., Reid, I. D., Molton, N. D. et Stasse, O. (2007). MonoSLAM : real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–67.

[Decrouez *et al.*, 2012a] Decrouez, M., Dupont, R., Gaspard, F. et Crowley, J. L. (2012a). Automatic object detection for modeling indoor environments. *In VISAPP*.

[Decrouez *et al.*, 2012b] Decrouez, M., Dupont, R., Gaspard, F. et Crowley, J. L. (2012b). Détection automatique d'objets pour la localisation en milieu intérieur. *In RFIA*.

[Decrouez *et al.*, 2012c] Decrouez, M., Dupont, R., Gaspard, F. et Crowley, J. L. (2012c). Extracting planar structures efficiently with revisited BetaSAC. *In ICPR*.

[Decrouez *et al.*, 2011] Decrouez, M., Dupont, R., Gaspard, F., Devernay, F. et Crowley, J. L. (2011). Modélisation explicite des objets et de l'environnement en combinant les approches topologique et métrique pour la localisation. *In ORASIS 2011 - 13e Congrès des jeunes chercheurs en vision par ordinateur*.

[Durrant-Whyte, 1987] Durrant-Whyte, H. (1987). Uncertain geometry in robotics. *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 4, pages 851–856.

[Durrant-Whyte et Bailey, 2006] Durrant-Whyte, H. et Bailey, T. (2006). Simultaneous localisation and mapping (slam) : Part I the essential algorithms. *IEEE Robotics and automation magazine*.

[Eade et Drummond, 2006] Eade, E. et Drummond, T. (2006). Scalable Monocular SLAM. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:469–476.

[Elfes, 1990] Elfes, A. (1990). Autonomous robot vehicles. chapitre Sonar-based real-world mapping and navigation, pages 233–249.

[Fischler et Bolles, 1981] Fischler, M. A. et Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

[Fraundorfer *et al.*, 2007] Fraundorfer, F., Engels, C. et Nistér, D. (2007). Topological mapping, localization and navigation using image collections. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3872–3877.

[Fukunaga et Hostetler, 1975] Fukunaga, K. et Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21:32–40.

[Guoshen Yu, Jean-Michel Morel, 2011] Guoshen Yu, Jean-Michel Morel (2011). ASIFT : An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*.

[Haralick *et al.*, 1994] Haralick, R. M., Lee, C.-N., Ottenberg, K. et Nölle, M. (1994). Review and analysis of solutions of the three point perspective pose estimation problem. *International Journal of Computer Vision*, 13(3):331–356.

[Harris et Stephens, 1988] Harris, C. et Stephens, M. (1988). A combined corner and edge detector. *In Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

[Hartley, 1997] Hartley, R. I. (1997). In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

[Hartley et Zisserman, 2000] Hartley, R. I. et Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*.

[Herbst *et al.*, 2011] Herbst, E., Henry, P., Ren, X. et Fox, D. (2011). Toward object discovery and modeling via 3-d scene comparison. *In ICRA*, pages 2623–2629.

[Ho et Newman, 2007] Ho, K. L. et Newman, P. (2007). Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286.

[Isack et Boykov, 2012] Isack, H. N. et Boykov, Y. (2012). Energy-based geometric multi-model fitting. *International Journal of Computer Vision*, 97(2):123–147.

[Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

[Kim *et al.*, 2010] Kim, K., Lepetit, V. et Woo, W. (2010). Keyframe-based Modeling and Tracking of Multiple 3D Objects. *In Proceeding of the 9th IEEE International Symposium on Mixed and Augmented Reality*.

[Klein et Murray, 2007] Klein, G. et Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. *In IEEE International Symposium on Mixed and Augmented Reality*.

[Kuipers et Byun, 1991] Kuipers, B. et Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of robotics and autonomous systems*, 8:47–63.

[Kundu *et al.*, 2010] Kundu, A., Krishna, K. M. et Jawahar, C. V. (2010). Real-time motion segmentation based multibody visual slam. *In ICVGIP*, pages 251–258.

[Lepetit *et al.*, 2009] Lepetit, V., Moreno-Noguer, F. et Fua, P. (2009). EPnP : An accurate O(n) solution to the PnP problem. *International Journal on Computer Vision*, 81(2):155–166.

[Levenberg, 1944] Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168.

[Lin et Wang, 2010] Lin, K.-H. et Wang, C.-C. (2010). Stereo-based simultaneous localization, mapping and moving object tracking. *In IROS*, pages 3975–3980.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110.

[Marquez-Gamez et Devy, 2012] Marquez-Gamez, D. et Devy, M. (2012). SLAM visuel avec détection et suivi d'objets mobiles par une approche de segmentation/classification. *In RFIA*.

[Mason *et al.*, 2012] Mason, J., Marthi, B. et Parr, R. (2012). Object Disappearance for Object Discovery. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[Matas *et al.*, 2002] Matas, J., Chum, O., Urban, M. et Pajdla, T. (2002). Robust wide-baseline stereo from maximally stable extremal regions. *In Proceedings of British Machine Vision Conference*, 1:384–393.

[Meger *et al.*, 2007] Meger, D., Forssén, P.-E., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J. J., Lowe, D. G. et Dow, B. (2007).

Curious george : An attentive semantic robot. *In IROS 2007 Workshop : From sensors to human spatial concepts.*

[Méler *et al.*, 2010] Méler, A., Decrouez, M. et Crowley, J. (2010). Betasac : A new conditional sampling for ransac. *In Proceedings of the British Machine Vision Conference.*

[Migliore *et al.*, 2009] Migliore, D., Rigamonti, R., Marzorati, D., Matteucci, M. et Sorrenti, D. G. (2009). Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. *In Proceedings of International workshop on Safe navigation in open and dynamic environments application to autonomous vehicles.*

[Mikolajczyk, 2004] Mikolajczyk, K. (2004). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60:63–86.

[Modayil et Kuipers, 2004] Modayil, J. et Kuipers, B. (2004). Towards bootstrap learning for object discovery.

[Moravec, 1988] Moravec, H. (1988). Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74.

[Mouragnon *et al.*, 2006] Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. et Sayd, P. (2006). Monocular Vision Based SLAM for Mobile Robots.

[Moutarlier et Chatila, 1989] Moutarlier, P. et Chatila, R. (1989). Stochastic multisensory data fusion for mobile robot location and environment modeling. *International Symposium on Robotics Research.*

[Ni *et al.*, 2009] Ni, K., Jin, H. et Dellaert, F. (2009). Groupsac : Efficient consensus in the presence of groupings. *In Proceedings of the IEEE 12th International Conference on Computer Vision.*

[Nistér, 2004] Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence.*

[Nister et Stewenius, 2006] Nister, D. et Stewenius, H. (2006). Scalable Recognition with a Vocabulary Tree. *In Proceedings of CVPR.*

[Reitmayr *et al.*, 2007] Reitmayr, G., Eade, E. et Drummond, T. (2007). Semi-automatic annotations in unknown environments. *In Proceedings of ISMAR 2007.*

[Russell *et al.*, 2006] Russell, B. C., Efros, A. A., Sivic, J., Freeman, W. T. et Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. *In Proceedings of CVPR.*

[Schindler et Suter, 2006] Schindler, K. et Suter, S. (2006). Two-view multibody structure-and-motion with outliers through model selection. *In Proceedings of PAMI.*

[Schmid *et al.*, 2000] Schmid, C., Mohr, R. et Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172.

[Singh et Ahuja, 2003] Singh, M. et Ahuja, N. (2003). Regression based bandwidth selection for segmentation using parzen windows. *In Proceedings of the IEEE International Conference on Computer Vision*, pages 2–9.

[Sivic et Zisserman, 2003] Sivic, J. et Zisserman, A. (2003). Video Google : a text retrieval approach to object matching in videos. *International Conference on Computer Vision.*

[Sivic et Zisserman, 2009] Sivic, J. et Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):591–606.

[Smith et Cheeseman, 1986] Smith, R. C. et Cheeseman, P. (1986). On the representation and estimation of spatial uncertainly. *The International Journal of Robotics Research*, 5(4):56–68.

[Southey et Figures, 2006] Southey, T. et Figures, L. O. (2006). Object discovery through motion, appearance and shape. Rapport technique, In AAAI Workshop on Cognitive Robotics.

[Strasdat *et al.*, 2010] Strasdat, H., Montiel, J. M. M. et Davison, A. J. (2010). Real-time monocular slam : Why filter ? *In ICRA*, pages 2657–2664.

[Subbarao et Meer, 2006] Subbarao, R. et Meer, P. (2006). Nonlinear mean shift for clustering over analytic manifolds. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1168–1175.

[Tardif *et al.*, 2008] Tardif, J.-P., Pavlidis, Y. et Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omnidirectional camera. *In IROS*, pages 2531–2538.

[Toldo et Fusiello, 2008] Toldo, R. et Fusiello, A. (2008). Robust multiple structures estimation with j-linkage. *In ECCV*.

[Torr et Zisserman, 2000] Torr, P. H. S. et Zisserman, A. (2000). MLESAC : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156.

[Triggs *et al.*, 2000] Triggs, B., McLauchlan, P. F., Hartley, R. I. et Fitzgibbon, A. W. (2000). Bundle adjustment - a modern synthesis. *In Proceedings of the International Workshop on Vision Algorithms : Theory and Practice*, pages 298–372.

[Walther *et al.*, 2004] Walther, D., Rutishauser, U., Koch, C. et Perona, P. (2004). On the usefulness of attention for object recognition. *In Workshop on Attention and Performance in Computational Vision at ECCV*, pages 96–103.

[Wang *et al.*, 2004] Wang, C., Thorpe, C., Hebert, M., Thrun, S. et Durrant-whyte, H. (2004). Simultaneous localization, mapping and moving object tracking. Rapport technique, International Journal of Robotics Research.

[Wang et Suter, 2004] Wang, H. et Suter, D. (2004). MDPE : A very robust estimator for model fitting and range image segmentation. *International Journal of Computer Vision*.

[Wangsiripitak et Murray, 2009] Wangsiripitak, S. et Murray, D. W. (2009). Avoiding moving outliers in visual slam by tracking moving objects. *In Proceedings of the 2009 IEEE international conference on Robotics and Automation*, pages 705–710.

[Zhang et Kosecka, 2006] Zhang, W. et Kosecka, J. (2006). Nonparametric estimation of multiple structures with outliers. *In Workshop on Dynamic Vision, European Conference on Computer Vision 2006*.

[Zuliani *et al.*, 2005] Zuliani, M., Kenney, C. S. et Manjunath, B. S. (2005). The multiransac algorithm and its application to detect planar homographies. *In IEEE International Conference on Image Processing*.